HABILITATION
A DIRIGER DES RECHERCHES

présentée par

William Puech

UNIVERSITÉ MONTPELLIER II

Spécialité : Traitement des Images

TRAITEMENT D'IMAGES À DISTANCE ET SÉCURISÉ PAR CRYPTAGE ET INSERTION DE DONNÉES CACHÉES

PARTIE I

Date de soutenance : 14 **Décembre** 2005

Composition du Jury :

Jean-Marc Chassery	President
Atilla Baskurt	Rapporteur
Jean-Pierre Coquerez	Rapporteur
Jean-Pierre Guedon	Rapporteur
Jean-Claude Bajard	Examinateur
André Crosnier	Examinateur
Gérard Michaille	Invite
Michel Robert	Invite

HDR préparée au sein du Laboratoire **LIRMM, UMR CNRS 5506**

A Magali.
A nos filles et notre fils.

Remerciements

Je tiens à remercier Monsieur Jean-Marc Chassery, Directeur de Recherche CNRS au LIS à Grenoble, mon *maître spirituel*, pour me suivre dans mes recherches depuis plus de 12 ans. Merci Jean-Marc, je sais que tu as encore de nombreux conseils à me donner pour les années futures. Je serai toujours à ton écoute... Merci également pour l'honneur qu'il m'a fait en présidant ce jury.

Je remercie Monsieur Atilla Baskurt, Professeur à l'INSA de Lyon, pour s'intéresser à mon travail et à celui de mes doctorants, et pour avoir accepté la lourde charge d'être rapporteur de mon habilitation à diriger des recherches.

Je remercie Monsieur Jean-Pierre Cocquerez, Professeur à l'Université Technologique de Compiègne, pour avoir également accepté d'être rapporteur de mon habilitation à diriger des recherches. Suite à son passage à Nîmes en 2002, ses remarques pertinentes ont réussi à m'orienter dans la bonne direction. Merci également pour avoir su me faire patienter.

La troisième personne qui a acceptée d'être rapporteur de mon habilitation à diriger des recherches est Monsieur Jeanpierre Guedon, Professeur à Polytech Nantes. Jeanpierre, en 2003, un certain Vincent Ricordel t'a proposé de me rencontrer afin de mettre en place un projet commun. Depuis nous sommes obligés de nous voir régulièrement et à partir de Janvier 2006, c'est parti pour 3 années de collaboration forte.

J'adresse également mes remerciements à Messieurs Jean-Claude Bajard et André Crosnier, Professeurs au LIRMM, à l'Université Montpellier II. Mes chers Directeurs de doctorants, j'espère que cette habilitation à diriger des recherches ne changera rien entre nous et que nous continuerons à co-diriger des étudiants en thèse.

Je tiens à remercier Monsieur Michel Robert, Professeur, Directeur du LIRMM et Directeur de l'Ecole Doctorale I2S, pour réussir à me canaliser en moyenne 3 fois par mois. Je sais que tu as encore du travail, je te souhaite bon courage. Je lui adresse ici toute ma reconnaissance.

Je tiens à remercier Monsieur Etienne Dombre, Directeur de Recherche CNRS et Monsieur Michel Habib, Professeur, ancien directeur du LIRMM pour m'avoir soutenu et avoir réussi à trouver un accord afin que mon passage du CEM2 au LIRMM se passe le mieux possible. J'en profite également pour remercier Monsieur Daniel Gasquet, Directeur de Recherche CNRS et Directeur du CEM2.

Je tiens à remercier de tout coeur tous les étudiants avec qui j'ai eu l'honneur de travailler et en particulier mes doctorants. Pour le passé, merci Khalifa, Jean-Claude, Grégory, pour le présent, merci José, Philippe, Jean-Luc et Zahia et pour le futur proche, merci Khyzar, Sabrina et Antoine.

Je ne saurais oublier de remercier tous les collègues de l'équipe ICAR du LIRMM qui m'ont accordé leur confiance afin de réunir nos forces et de faire chemin ensemble. Je

tiens à remercier en particulier Olivier Strauss, avec qui je fais équipe pour de nombreuses *affaires*. Merci Marie-José, André, Christophe, Frédéric, Jean et merci aux nouveaux, à savoir Marc et Sébastien pour leur dynamisme et pour le futur.

Je tiens enfin à exprimer ma sympathie à toutes les personnes que j'ai pu côtoyer au sein des différents laboratoires et universités que j'ai fréquentés. Merci à tous les techniciens, secrétaires et ingénieurs pour leur disponibilité permanente et pour réussir et accepter de dialoguer avec des personnes d'un *univers particulier*.

Je remercie Gille Gesquières, Maître de Conférences à l'IUT d'Arles, collègue et ami, pour avoir bien voulu prendre quelques jours afin de relire mon document. Je remercie également un autre collègue qui est aussi un ami, à savoir Jean Triboulet avec qui j'ai partagé mon désespoir presque deux années durant.

Je tiens également à remercier mes collègues du CUFR de Nîmes pour le travail pédagogique et administratif que nous menons ensemble.

Enfin pour terminer, je tiens à remercier mes trois enfants pour accepter mon travail en week-end ou mes rentrées tardives en semaine. Promis, je vais me contrôler.

Pensée particulière à Magali.

La technique consistant à savoir le plus de choses possible sur des sujets de plus en plus restreints aboutit en fin de compte à savoir tout sur rien. Avec ce système, il n'y a plus un savant, il y a des myriades de savants, qui ne sont utiles que groupés ; isolés, ils ne sont rien : ils savent tout ce qu'on peut savoir sur une minuscule aire d'investigation, mais ce qu'ils savent n'est en fin de compte qu'un boulon dans une immense machine dont, la plupart du temps, ils sont incapables de savoir ce qu'elle signifie et á quoi elle va servir.

Jean Giono, Les trois arbres de Palzem.

Table des matières

Introduction générale

Il y a 10 ans ma première publication paraissait au GRETSI 95 à Juan les Pins. Voila 8 ans que j'ai soutenu ma thèse en 1997 à Grenoble. Depuis la soutenance de ma thèse j'ai changé de thématiques de recherche, de statuts ainsi que de laboratoires. Dans ce manuscrit, nous allons donc voir ensemble comment ma thématique scientifique a régulièrement évolué, et je justifierai tout au long de ce mémoire la cohérence de cette mobilité scientifique. Celle-ci est fortement corrélée à ma mobilité géographique.

Mes activités de recherche ont débuté en 1991 au LIFIA à Grenoble dans le cadre du DEA Signal-Image-Parole, sous la direction de Radu Horaud. J'ai continué en thèse au TIMC à Grenoble sous la direction Jean-Marc Chassery. J'ai effectué durant mes années de thèse un séjour de 8 mois à l'Université de Thessalonique en Grèce, avec Ioannis Pitas. A Toulon, j'ai intégré le laboratoire MS dirigé par Pierre-Yves Arques. A mon arrivée à Montpellier, j'ai travaillé 2 ans dans le laboratoire CEM2 dirigé par Robert Alabedra. Je suis depuis 3 ans au LIRMM dirigé par Michel Robert.

Dans ce document, je présente principalement les résultats obtenus après ma soutenance de thèse à Grenoble en 1997 jusqu'à ce jour à Montpellier. A la suite de ma thèse, qui a eu lieu à Grenoble le 3 septembre 1997, j'ai été détaché à l'Université de Toulon et du Var sur un poste de PRAG (Professeur Agrégé) en physique appliquée. Pendant les trois années suivantes, de 1997 à 2000, j'ai continué à effectuer des recherches en collaboration avec un médecin spécialiste afin de développer un système de visualisation et de traitement d'images médicales à distance. Durant cette période j'ai co-encadré, un doctorant qui a travaillé sur des contours actifs géodésiques ainsi que trois étudiants en DEA.

En septembre 2000 j'ai été recruté à l'Université Montpellier II sur un poste de MCF 61 (Maître de Conférences) et j'ai poursuivi mes recherches sur le traitement d'images à distance tout en insistant sur la partie sécurisation du transfert. Un deuxième doctorant que j'ai co-encadré a travaillé sur l'application des méthodes de cryptage aux images. En 2002, un troisième co-encadrement de doctorant a démarré ses recherches sur l'insertion de données cachées sécurisée par des codes correcteurs d'erreurs appliquées aux images couleurs et robustes à la compression.

En 2003, j'ai rejoint le LIRMM tout en étant rattaché pour la partie enseignement au CUFR de Nîmes. Durant la même période, j'ai débuté le co-encadrement d'un ingénieur brésilien, détaché pendant trois ans dans le cadre d'une thèse sur la combinaison des algorithmes de cryptage et des méthodes d'insertion de données. En 2004, deux nouveaux doctorants rejoignaient également cette thématique sur les images sécurisées dans le cadre de leur thèse sur la protection d'objets 3D pour l'un et sur l'insertion de données cachées

dans des images robuste aux transformations géométriques discrètes pour l'autre. Depuis 2000, j'ai également encadré 7 étudiants en DEA.

En septembre 2005, démarre une thèse en cotutelle entre la France et l'Algérie avec une doctorante qui travaille sur la crypto-compression d'images médicales par ondelette.

Les travaux que je présente dans le cadre de mon habilitation à diriger des recherches sont donc le fruit de travaux que j'ai réalisé avec plus d'une quinzaine d'étudiants, doctorants ou étudiants de DEA, ainsi qu'avec des collègues chercheurs. De mon point de vue, la recherche est avant tout un travail d'équipe qui nécessite énormément de communication.

Depuis deux ans, j'ai la responsabilité de l'équipe **ICAR** (Image et Réalité virtuelle) du LIRMM. Cette équipe est composée de neuf personnes travaillant en Image, Signal, Vision, 3D, Informatique graphique et Réalité virtuelle.

Ce document est composé de deux parties. Dans la première partie je décris mes activités scientifiques, pédagogiques et administratives. Dans la deuxième partie de ce manuscrit, je détaille mes activités de recherche que j'ai décomposé en 5 chapitres allant du traitement et de la visualisation d'images à distance jusqu'au codage hybride associant cryptage, insertion de données cachées et compression. Je termine ce document par une conclusion et des perspectives de recherche sous forme de sujets de thèse.

Première partie

Descriptif des activités

Introduction

Dans cette première partie de mon document, après une description rapide, chapitre 2, de mon parcours de recherche depuis la soutenance de ma thèse jusqu'à mon intégration au LIRMM, je présente, chapitre 3, mes activités pédagogiques et administratives. Dans le chapitre 4, je présente la liste des doctorants et étudiants en DEA que j'ai co-encadré ou que j'encadre actuellement. Je décris, chapitre 5, mes participation à des réseaux scientifiques et à des séminaires en tant qu'invité. Je présente également, chapitre 6, les financements que j'ai obtenus par l'intermédiaire d'aides régionales, d'actions du CNRS et de contrats de collaborations avec des entreprises. J'ai tenu également, dans cette partie à vous présenter, chapitre 7, les projets en cours avec lesquels je compte continuer à développer mes activités scientifiques. Le chapitre 8 résume la seconde partie de ce document qui contient en détail mes activités de recherche. Enfin, chapitre 9, je propose une liste mes publications.

Chapitre 1

Curriculum Vitae

William PUECH , MCF 61ème section depuis le 01/09/2000
Né le 26/12/1967 (37 ans) à Nîmes, Marié, 3 enfants.

Enseignement : Centre Universitaire de Formation et de Recherche de Nîmes (CUFRN), Place Gabriel Péri, 30021 Nîmes Cedex 1, 0466279580.

Recherche : Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), UMR CNRS 5506, UMII.

Formation et parcours :

1992 : DEA Signal Image Parole de l'Institut National Polytechnique de Grenoble, obtenu en juin 1992, mention AB. Stage effectué au LIFIA en vision par ordinateur sous la direction de M. Horaud Radu.

1993 : Service national effectué en tant qu'aspirant puis sous-lieutenant.

1994 : Thèse 1ère année, *Localisation, reconstruction et mosaïque appliquées aux peintures sur cylindres généralisés à axe droit en vision monoculaire*, sous la direction de Jean-Marc Chassery, Laboratoire (TIMC-IMAG).

1995 : Capes Physique Appliquée préparé à l'IUFM et obtenu en juin 1995.

1996 : Séjour doctoral de 8 mois entre novembre 1995 et juin 1996. Validation de ma 2ème année de thèse.

1997 : Thèse en Signal-Image-Parole à l'Institut National Polytechnique de Grenoble, obtenue le 3 septembre 1997. *Localisation, reconstruction et mosaïque appliquées aux peintures sur cylindres généralisés à axe droit en vision monoculaire*, sous la direction de Jean-Marc Chassery, Laboratoire des Techniques de l'Imagerie, de la Modélisation et de la Cognition, Institut d'Informatique et de Mathématiques Appliquées de Grenoble (TIMC-IMAG).

1997 : Professeur stagiaire, certifié en Physique Appliquée en poste au lycée Dhuoda, à Nîmes (en parallèle de ma troisième année de thèse).

1997-2000 : PRAG à l'Université de Toulon et du Var, en poste au département Service et Réseaux de Communication de Saint-Raphaël.

1998 : Qualification aux sections CNU 27 et 61.

Depuis 2000 : MCF 61 en poste à l'UMII sur le site de Nîmes, puis rattaché au CUFR de Nîmes.

2003 : PEDR, obtention de la PEDR (Prime d'Encadrement Doctoral et de Recherche en septembre 2003.

Enseignements :

Informatique : programmation objet, algorithmique et informatique graphique.

Réseaux : transmission, architecture, application Client/Serveur et télécom.

Image : acquisition, traitement, codage et compression.

Recherche :

Traitement des images, vision par ordinateur et reconstruction 3D

Codage, compression et cryptage des images

Transfert sécurisé par insertion de données cachées

Encadrements :

Co-encadrements doctorants depuis 1999 : 3 doctorants soutenus et 3 co-encadrements en cours, plus 1 encadrement en co-tutelle.

Encadrements DEA : 13 étudiants en DEA soutenus (DEA Informatique de Grenoble et Montpellier, DEA Optique-Image-Signal de Toulon, DEA Automatique de Montpellier).

Publications :

Chapitre de livre : 1 chapitre de livre en cours de rédaction (Traité IC2, Hermès-Lavoisier)

Publications revues : 6 revues internationales + 3 soumises (PRL, PR, SPIC, IJIG, JIST).

Conférences internationales : 29 conférences internationales (principalement EUSIPCO, ICIP, ICAPR, WIAMIS, ICASSP et CGIV).

Conférences nationales : 19 conférences nationales (principalement GRETSI, CORESA et GBM).

Brevet : 1 brevet déposé par le CNRS en 2005 en collaboration avec une entreprise.

Premier prix du Collège des enseignants d'Informatique, Biomathématiques, Méthodes en épidémiologie, statistique section Biostatistique, Informatique médicale et Technologie de la Communication. Université Technologique de Compiègne, 6 Juin 2001.

Responsabilités et animations :

Module Image : de l'école doctorale I2S de l'UMII depuis 2002.

GDR ISIS : Participation à de nombreuses journées du GDR ISIS.

Equipe Projet ICAR (Image et Réalité virtuelle) : du LIRMM, responsable de cette équipe-projet depuis sa création en 2004.

Responsable des enseignements Informatique : de la licence Mathématiques-Informatique du CUFR de Nîmes.

Contrats et financement : 2 financements (Région et Université), 1 financement par une Action Spécifique du CNRS et 5 contrats avec entreprises (entre 6000 euros et 10000 euros par contrat).

Projet de l'ANR : le projet TSAR (Transfert Sécurisé d'images d'Art haute Résolution) a été retenu par l'ANR ARA SSIA pour la période 2006-2008 en partenariat avec le C2RMF, UMR CNRS (musée du Louvre), Paris.

Président de session : pour la conférence EUSIPCO 2004.

Lecteur : pour les revues PRL et AES et la conférence IWDW.

Expert de projets : pour l'ANVAR.

Chapitre 2

Parcours

2.1 Thèse en Signal-Image-Parole

Le 3 septembre 1997, j'ai soutenu ma thèse en Signal-Image-Parole à l'Institut National Polytechnique de Grenoble. J'ai effectué ma thèse sur la *Localisation, reconstruction et mosaïque appliquées aux peintures sur cylindres généralisés à axe droit en vision monoculaire*, sous la direction de Jean-Marc Chassery, Laboratoire des Techniques de l'Imagerie, de la Modélisation et de la Cognition, Institut d'Informatique et de Mathématiques Appliquées de Grenoble (TIMC-IMAG). Le président du jury était M. Jean-Louis Lacoume et les rapporteurs Mme Isabelle Magnin et M. Michel Dhome. Mme Annick Montanvert ainsi que Monsieur Ioannis Pitas faisaient également partie du jury en tant qu'examinateurs.

L'objectif de ces travaux concernent la localisation et la reconstruction de surfaces cylindriques sur lesquelles sont projetées ou plaquées des scènes que le capteur perçoit comme des images. Du fait d'une étude limitée à la vision monoculaire, l'utilisation de connaissances a priori est nécessaire. [Puech 97 RI]. Nous avons présenté deux méthodes de détection de la projection de l'axe d'un CGUD (Cylindre Généralisé Uniforme à axe Droit), puis décrit comment obtenir un deuxième axe dans l'image. La signification de ces deux axes a été interprétée pour décrire une méthode de reconstruction 3D de CGUD pouvant être étendue aux CGHD (Cylindre Généralisé Homogène à axe Droit) de sections fermées circulaires ou elliptiques, ou de sections ouvertes paraboliques ou elliptiques [Puech 97 b CI]. Nous avons démontré que l'évolution des courbures des ellipses dans l'image, projections de sections du CGUD de sections circulaires, est fonction linéaire de l'altitude de la section dans l'espace 3D [Puech 98 CI]. Nous avons étendu ces travaux aux problèmes de mosaïques de surfaces cylindriques [Puech 01 a RI]. J'ai effectué en 1996 un séjour de six mois dans le laboratoire Computer Vision and Image Processing du Professeur I.

Pitas (Aristotle University of Thessaloniki, Grèce) dans le cadre d'un projet EURODOC intitulé "Redressement de Peintures Murales sur Surfaces Courbes". Durant ces années de thèse, j'ai encadré avec J.M. Chassery un étudiant en DEA Informatique : P. Schott, "Identification de la Géométrie de Surfaces Courbes en Vision Monoculaire", 1995.

2.2 PRAG physique appliquée à l'Université de Toulon

De septembre 1997 à août 2000 j'ai occupé un poste PRAG (Professeur Agrégé) en physique appliquée, en tant que certifié en physique appliquée, à l'Université de Toulon et du Var. J'ai intégré le laboratoire Modélisation et Signal (MS), dirigé par Pierre-Yves Arques, laboratoire de l'Université de Toulon et du Var.

Durant ces trois années, j'ai travaillé sur des *Méthodes automatiques de détections de contours pour la reconstruction 3D d'organes anatomiques*. J'ai mis en place une collaboration avec le Centre Hospitalier Intercommunal de Fréjus, avec le Docteur radiologue Guy Passail. L'objectif de ces travaux était de développer des méthodes de reconstruction 3D semi-automatique de l'aorte à partir de coupes d'images tomodensitométriques. Dans la première coupe, des contours actifs géodésiques permettaient de détecter le premier contour. A partir de ce contour obtenu dans la première coupe, la méthode proposée permettait une propagation dans les coupes voisines.

Nous avons constaté que certains problèmes subsistaient quand à la détection de contours. Nous avons alors proposé des techniques d'amélioration de ces méthodes en réalisant semi-automatiquement l'extraction d'une seule structure anatomique ciblée en mettant en oeuvre un modèle de contours actifs. Les images obtenues par reconstruction 3D constituent un apport diagnostic important pour l'étude des pathologies de l'aorte [Puech 99 CN, Puech 00 b CI]. Sur cette thématique j'ai encadré trois étudiants en DEA Optique, Image et Signal, Lab. MS, dont un avec V. Ricordel ; S. Nicolay, "Détection de Contours dans une Séquence d'Images par Contours Actifs. Application à l'imagerie médicale", Juin 1999 ; G. Ledanff, "Mise en place de techniques de détection de contours dans une image médicale par contours actifs géodésiques", Juin 2000 ; J. Michelis, "Propagation automatique d'un contour dans une séquence d'images médicales en vue d'une reconstruction 3D", Juin 2000.

J'ai également co-encadré les travaux de thèse de Djemal Khalifa, dirigé par B. Rossetto, Laboratoire d'Optique Appliquée, Université de Toulon et du Var, "Segmentation par contour actif et suivi automatique d'un objet dans une séquence d'images", 16 Décembre

2002. Jury : B. Rossetto, P. Réfregier, J-M. Chassery, B. Jouvencel, W. Puech. Ce travail présentait des algorithmes de segmentation par contours actifs déformables permettant le suivi automatique d'un objet sur une séquence d'images temporelle ou spatiale. La formulation par courbes de niveaux de ces modèles permettait une gestion automatique des changements de topologie. En appliquant nos méthodes sur une séquence d'images tomodensitométriques de l'aorte abdominale, et après avoir détecté les contours sur chacune des images de la séquence, nous avons pu réaliser une reconstruction 3D de cet organe [Djemal 02 a CI, Djemal 02 b RI].

De 1997 à 2000 nous avons également mis en place des applications Client/Serveur permettant de faciliter l'accès à une base de données composée d'images de taille importante (format DICOM Digital Image Communication). Cette application Client/Serveur permettait la visualisation d'images issues de différents appareils de radiologie.

2.3 MCF Signal-Image, Université Montpellier II, CEM2

En septembre 2000 j'ai obtenu un poste de Maître de Conférences en 61ème section (Signal-Image) à l'Université Montpellier II. Jusqu'en août 2003, j'ai été rattaché au Centre d'Electronique et de Microoptoélectronique de Montpellier (CEM2), dirigé par R. Alabedra, Université Montpellier II. Ma thématique de recherche s'est orientée vers le *Transfert sécurisé d'images par cryptage et tatouage pour la télé-application.*

Nous avons développé des techniques de tatouage et de cryptage d'images pour le transfert sécurisé. Les applications vont de l'imagerie médicale à la sécurité routière : développement de techniques combinant la cryptographie et le tatouage d'images, problème de la sécurisation dans l'utilisation des réseaux informatiques pour la transmission d'informations médicales, méthodes de cryptage d'images médicales basé sur des algorithmes de chiffrement par flux, utilisation de l'algorithme RSA en regroupant les pixels d'une image par blocs pour augmenter la taille des clefs de cryptage.

Nos recherches se sont appuyées sur des développements de techniques combinant la cryptographie et le tatouage d'images [Puech 01 d CN, Puech 01 e CN]. La cryptographie permet de rendre une information illisible pendant le transfert, alors que le tatouage permet d'insérer un message de manière invisible et indélébile dans les données.

Les recherches effectuées dans le cadre de la thèse de Jean-Claude Borie, intitulée "Sécurisation d'images par cryptage : applications aux images médicales", dirigée par Michel Dumas et que j'ai co-encadrée, intéressent fortement les médecins spécialistes en

imagerie. Concernant la cryptographie, l'utilisation des réseaux informatiques pour la transmission d'informations médicales pose le problème de la sécurisation des données. Dans une de nos premières méthodes de cryptage d'images médicales, nous avons utilisé un algorithme dérivé de celui de Vigenère [Puech 01 f CN]. Nous nous sommes ensuite appuyés sur l'algorithme RSA en regroupant les pixels d'une image par blocs afin d'augmenter la taille des clefs de cryptage [Borie 02a CI, Borie 02b CI]. Concernant le tatouage, notre objectif etait de récupérer après le transfert un message long contenu dans l'image. Le contenu du message long, dans le cadre d'une application médicale concerne le nom du patient, les propriétés de l'examen, les séries de l'examen, les paramètres des images et le chemin exact vers ces images.

Concernant les applications du type sécurité routière ou télésurveillance, nous cherchons à stocker des informations telles que date, heure et lieu d'acquisition de l'image, ainsi que le numéro et la référence de la caméra, mais aussi des informations extraites dans l'image. Par conséquent, si l'image subie une modification durant le transfert, nous sommes alors capables de redonner les informations originales contenues dans l'image. Les recherches sont actuellement effectuées dans ce domaine dans le cadre de la thèse de Grégory Lo-Varco, intitulée "Transfert sécurisé d'images couleurs par insertion de données cachées et codes correcteurs d'erreurs", dirigée par Michel Dumas et que j'ai co-encadré [Lovarco 03 CN]. Depuis 2000, ces thématiques sont développées en partie avec la collaboration de P. Montesinos, du LGI2P de l'Ecole des Mines d'Ales, pour le traitement d'images couleur robuste à la compression [Puech 02 CI].

2.4 MCF Signal-Image, CUFR Nîmes, LIRMM

Depuis septembre 2003, je suis rattaché au Centre Universitaire de Formation et de Recherche de Nîmes et au Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), Université Montpellier II, dirigé par M. Habib jusqu'en décembre 2004, puis par Michel Robert. Mes recherches se sont alors orientées vers les *Combinaisons d'algorithmes d'insertion de données et de cryptage appliquées aux images et aux objets 3D.*

J'ai intégré le département Robotique du LIRMM et je collabore avec des chercheurs du département Informatique. Ma thématique de recherche s'oriente vers le développement de méthodes combinant le cryptage et l'insertion de données pour la sécurisation d'images pendant et après le transfert. Un des objectifs à atteindre est de proposer un cryptage

d'images multi-niveaux afin que la même image soit accessible à différents niveaux de résolution en fonction des droits attribués à l'utilisateur. Ces travaux qui intéressent le musée du Louvre dans le cadre d'une application permettent de sécuriser la haute résolution de peintures numériques. Un autre objectif est d'augmenter le contenu d'information (d'où insertion de données plutôt que tatouage) dans une image. Nous nous orientons également vers des méthodes autonomes de cryptage d'images en combinant les propriétés des algorithmes symétriques et asymétriques. Enfin pour certaines applications, comme l'imagerie médicale, nous développons des méthodes d'insertion de données totalement réversibles afin de restituer parfaitement l'image originale. Actuellement mes recherches s'intègrent complètement dans le projet ICAR du LIRMM que je dirige. Depuis mon intégration au LIRMM, 4 thèses que je co-encadre ont débuté. José Rodrigues a commencé sa thèse en 2003 intitulée "Crypto-tatouage et crypto-compression pour le transfert et l'archivage d'image", sous la direction de Jean-Claude Bajard. En 2004, Philippe Amat et Jean-Luc Toutant ont également débuté leur thèse intitulée respectivement "Digitalisation sécurisée d'objets 3D : application aux formes et aux lignes de styles de chaussures", sous la direction d'André Crosnier et "Tatouage sécurisé robuste aux rotations discrètes", sous la direction de Jean-Claude Bajard. Enfin, en 2005, Mlle Zahia Brahimi a commencé sa thèse en co-tutelle intitulée "Cryptage partiel d'images médicales", sous la direction de H. Bessalah et de moi-même.

Chapitre 3

Responsabilités pédagogiques et administratives

3.1 Enseignements et responsabilités pédagogiques

3.1.1 Enseignements

– Durant ma thèse et de par ma formation d'enseignant du secondaire j'ai effectué divers enseignements à Grenoble, à Toulon et à Nîmes :

1996-1997 : **192 h**[1] en tant que professeur Certifié en Physique et Electricité Appliquée au Lycée technique Régional Dhuoda à Nîmes, Gard. Classes de première et terminale Génie Electronique (F2). Enseignement en Electronique et Physique Appliquée.

1997 : 55 h de vacations en Technologie-Informatique : programmation en C et en assembleur 68000 en première année à l'IUT d'Informatique, Université P. Mendes France de Grenoble.

1997 : 24 h de vacations en Réseaux et Programmation JAVA avec application Client/Serveur, à l'IUT de Services et Réseaux de Communications (SRC) à l'Isle d'Abeau, Université J. Fourier de Grenoble.

– Durant mes trois années passées à l'IUT de Services et Réseaux de Communications (SRC) à Saint Raphaël, Université de Toulon et du Var, j'ai enseigné en IUT, mais également en maîtrise et en DESS :

1997-2000 : **384 h/an** d'enseignement en tant que PRAG en SRC : algorithmique, programmation orientée objet C++, Java, JavaScript, réseau, mise en ligne de

1. **en gras : heures effectuées dans le cadre de mon service**

sites, application Client/Serveur; scripts CGI, perl, php, base de données en ligne.

1997-2000 : 45 h de Réseaux télécoms, Maîtrise EEA : programmation Internet, développement des réseaux télécoms, application Client/Serveur, Université de Toulon et du Var.

1998-1999 : Mise en place d'un module de réseaux avec application Client/Serveur pour la Maîtrise Informatique, Université de Toulon et du Var.

1998-2000 : 55 h d'Optique et de Traitement du Signal pour le Traitement des Images et Vision par Ordinateur, les capteurs optiques et les modèles géométriques de caméras. Diplôme d'Etude Supérieure Universitaire (DESU) Services et Multimédia en Ligne, (1ère et 2ème année), Université de Toulon et du Var.

1999-2000 : 15 h d'architecture TCP/IP, DU Réseaux : les couches, adresses IP, routages, protocoles et les services réseaux, Université de Toulon et du Var.

1999-2000 : 15 h d'Optique-Vision 3D, DESS Ingénierie Marine : formation d'une image, radiométrie, photométrie, colorimétrie, capteurs d'acquisitions, vision 3D, calibrage; modélisation d'une caméra, repères, Université de Toulon et du Var.

– Depuis que je suis maître de conférences (2000) j'effectue la totalité de mon service sur le site de Nîmes, j'ai également en charge des enseignements à l'IUT d'Arles dans le département d'Informatique, option Imagerie et en DEA d'Informatique de l'Université Montpellier II :

2000-2002 : **192 h/an** en Introduction aux architectures de communication, OSI, TCP/IP, Programmation Java, Application Client/Serveur, Théorie de télécommunications, Traitement des Images, IUP MIC (Métiers de l'Information de la Communication), Nîmes, Université Montpellier II

2000-2001 : 12 h/an, Programmation Java, l'IUT Toulon, SRC, Université de Toulon et du Var

2000-2002 : 12 h/an, Introduction de la chaîne d'acquisition au traitement des images, Licence Arts Appliqués, Nîmes, Université Montpellier III

200-2002 : 30 h, Informatique, Bureautique, Option DEUG MIAS, SM et SVT, Nîmes, Université Montpellier II

2000-2001 : 24 h, Système d'exploitation Linux, Formation continue IRISM, IUT de Nîmes, Université Montpellier II

2000-2001 : 20 h, Informatique, Programmation Scheme, DEUG MIAS, Nîmes, Université Montpellier II

2002 : 30 h d'informatique programmation orientée objet, IUT d'Informatique option Imagerie sur le site d'Arles.

2003-2005 : 20 h/an de Traitement des images : filtrage et segmentation, IUT d'Arles dans le département d'Informatique, option Imagerie.

2003 : 20 h de cryptographie à l'institut EERIE de l'EMA (Ecole des Mines d'Ales) sur le site de Nîmes.

2003-2005 : **180 h/an** en Licence (L1, L2, L3) Math-Info du CUFR de Nîmes. En L1 : Introduction à l'algorithmique et aux systèmes. En L2 : Programmation orientée objet. En L3 : Réseaux et compression.

2003-2005 : **12 h/an** d'imagerie en Licence Biotechnologie (L3), CUFR de Nîmes.

2003-2005 : 10 h/an en Codage, compression et protection des images en DEA d'Informatique de l'Université Montpellier II.

3.1.2 Responsabilités pédagogiques

- Responsable pédagogique de la première année de l'IUP MIC (Métiers de l'Information et de la Communication), (2000-2004), CUFR Nîmes.
- Responsable pédagogique de l'informatique en Licence Mathématiques et Informatique (L1, L2, L3), (2000-2005), CUFR Nîmes. Actuellement je pilote pour l'informatique une équipe pédagogique constituée de 3 permanents, 3 moniteurs et une dizaine d'intervenants extérieurs pour environ un millier d'heures en informatique.
- Co-responsable d'un module de traitement d'images de l'Ecole Doctorale I2S (Information Structures Systèmes) de l'UMII. Ce module, ouvert aux doctorants depuis 2003, est suivi par 20 doctorants par an. En effet, depuis 3 ans, le directeur de l'école doctorale I2S, Michel Robert, m'a donné la co-responsabilité de développer des formations en "Signal-Image-Vision". Cette responsabilité s'est concrétisée par l'existence d'un module de l'école doctorale. Dans le cadre de ce module, des chercheurs connus au niveau national et international en Image interviennent régulièrement pour des cours-séminaires de 3h suivis par 20 à 40 personnes, étudiants et chercheurs.
- Tuteur de moniteurs depuis 2001 : 3 en 61 et 2 en 27. Depuis mon arrivée sur le site de Nîmes, j'ai eu à encadrer les enseignements de 5 moniteurs qui interviennent principalement en informatique et en traitement de l'information.

3.2 Responsabilités administratives

- Membre élu du Conseil Scientifique du Centre Universitaire de Formation et de Recherche de Nîmes.
- Membre nommé exterieur de la Commission de Spécialistes 61ème section de l'UMII (Université Montpellier II).
- Membre titulaire de la Commission de Spécialistes Math-Info-Physique du CUFR de Nîmes.
- Président de jurys de Bac S, 2003 à Nîmes, 2004 et 2005 à Alès, Gard.
- Membre du jury de DEA SYAM (Systèmes Automatique et Microélectronique) de l'UMII.
- Président de jury de la licence Math-Info du CUFR de Nîmes (2003-2005).

Chapitre 4

Co-encadrement de thèses et de DEA

J'ai co-encadré 3 doctorants dont un à Toulon et 2 à Montpellier-Nîmes. Je co-encadre actuellement 4 doctorants dont une doctorante en co-tutelle avec le CDTA d'Alger en Algérie. J'ai encadré également 13 étudiants durant leur stage de DEA.

4.1 Thèses soutenues

- K. Djemal. 16 décembre 2002. *Segmentation par contour actif et propagation automatique dans une séquence d'images*. Thèse Université de Toulon et du Var, LOA. Directeur : B. Rossetto, Co-encadrant : W. Puech (60%), Rapporteurs : B. Jouvencel et JM. Chassery, Membre du jury : P. Réfregier.
- JC. Borie. 8 décembre 2004. *Sécurisation d'images par cryptage : application aux images médicales*. Thèse Université Montpellier II, CEM2, Directeur : M. Dumas, Co-encadrant : W. Puech (90%), Rapporteurs : J.P. Guedon et C. Roux, Membres du jury : D. Gasquet et P. Falgayrettes.
- G. Lo-Varco. 26 septembre 2005. *Transfert sécurisé d'images couleurs par insertion de données cachées et codes correcteurs d'erreurs*. Directeur : M. Dumas, Co-encadrant : W. Puech (95%), CEM2, Université Montpellier II. Allocataire. Rapporteurs : J.M. Chassery et A. Baskurt, Membres du jury : D. Gasquet et M. Robert.

4.2 Thèses en cours

- J.M. Rodrigues. *Crypto-tatouage et crypto-compression pour le transfert et l'archivage d'image*. JC. Bajard, C. Fiorio et W. Puech (80%). LIRMM, Univ. Mtp II. Fin septembre 2006. Ingénieur informatique, Université de Ceara, Brésil.

- J.L. Toutant. *Tatouage sécurisé robuste aux rotations discrètes*. JC. Bajard, C. Fiorio et W. Puech (45%). LIRMM, Univ. Mtp II. Fin septembre 2007. Allocataire.
- Ph. Amat. *Digitalisation sécurisée d'objets 3D : application aux formes et aux lignes de styles de chaussures*. A. Crosnier et W. Puech (80%). LIRMM, Univ. Mtp II. Fin septembre 2007. Bourse cifre.
- Z. Brahimi. *Cryptage partiel d'images médicales*. W. Puech. et H. Bessalah, Thèse en co-tutelle : CDTA (Centre de Développement des Technologies Avancées) d'Alger en Algérie et le LIRMM, Univ. Mtp II. Fin septembre 2008.

4.3 DEA soutenus

- P. Schott. *Identification de la géométrie de surfaces courbes en vision monoculaire*. DEA Informatique, TIMC-IMAG, J.M. Chassery et W. Puech, Grenoble, Juin 1995
- S. Nicolay. *Détection de Contours dans une Séquence d'Images par Contours Actifs. Application à l'imagerie médicale*. DEA Optique, Image et Signal, Lab. MS, W. Puech, Toulon, Juin 1999.
- G. Ledanff. *Mise en place de techniques de détection de contours dans une image médicale par contours actifs géodésiques*. DEA Optique, Image et Signal, Lab. MS, W. Puech, Toulon, Juin 2000.
- J. Michelis. *Propagation automatique d'un contour dans une séquence d'images médicales en vue d'une reconstruction 3D*. DEA Optique, Image et Signal, Lab. MS, W. Puech, V. Ricordel, Toulon, Juin 2000.
- G. Lo-Varco. *Tatouage d'images pour la sécurité routière*. DEA Electronique, Composants et Systèmes, Lab. CEM2, W. Puech, Montpellier, Juin 2002.
- S. Piat. *Combinaison de méthodes de cryptage d'images*. DESS Informatique Réseaux Image, Université de Reims, W. Puech, Septembre 2003.
- G. Benoît. *Parcours de blocs pour l'optimisation d'algorithmes de crypto-compression*. DEA SYAM, LIRMM, W. Puech, Montpellier, Juin 2004.
- P. Amat. *Extraction d'information haute résolution dans l'image pour le tatouage*. DEA SYAM, LIRMM, W. Puech, Montpellier, Juin 2004.
- JL. Toutant. *Tatouage d'une signature électronique dans une signature visuelle*. DEA Informatique, LIRMM, C. Fiorio et W. Puech, Montpellier, Juin 2004.
- S. Martineau. *Crypto-compression d'images par quadtree et décomposition en plans binaires*. DEA SYAM, LIRMM, C. Fiorio et W. Puech, Montpellier, Juin 2005.

– O. Léger. *Crypto-compression d'images par octree*. DEA Informatique, LIRMM, W. Puech, Montpellier, Juin 2005.

– D. Falguère. *Analyse de la robustesse au bruit d'un système de crypto-compression d'images*. DEA SYAM, IMERIR Perpignan, Montpellier, Septembre 2005.

– A. Martin. *Insertion d'un modèle numérique de terrain dans une carte de textures multirésolution*. DEA SYAM, IMERIR Perpignan, Montpellier, Septembre 2005.

Chapitre 5

Participation à des réseaux et séminaires invités

5.1 Participation à des réseaux

- Lecteur pour la revue Pattern Recognition Letters, Elsevier Science.
- Lecteur pour la revue Advances in Engineering Software, Science Direct.
- Lecteur pour la conférence IWDW, International Workshop of Digital Watermarking.
- Membre correspondant du GDR ISIS (Information, Signal, Image et Vision) pour le LIRMM.
- Membre du GDR STIC-Santé.
- Membre du GDR ALP (Algorithmique, Langage et Programmation)
- Co-animateur du Club Image, regroupant des spécialistes de traitement d'images en Languedoc Roussillon, Ecole Doctorale I2S.
- Expert ANVAR (Agence Nationale de Valorisation de la Recherche). En 5 ans j'ai effectué 5 expertises pour ANVAR PACA (Provence, Alpes et Côte d'Azur) et ANVAR Poitou-Charentes.
- Président de session (chairman) pour la conférence EUSIPCO 2004 (European Signal and Image Processing Conference), Vienne, Autriche. SESSION TuePmOR6: Multidimensional Systems and Signal Processing (Lecture).
- Membre de l'Action Spécifique CNRS "Contenu Sécurisé et Tatouage"
- Prix du jury du CIMES : Collège des enseignants d'Informatique, Biomathématiques, Méthodes en épidémiologie, statistique section Biostatistique, Informatique médicale et Technologie de la Communication. Université Technologique de Compiègne, 6 Juin 2001.

- Invité à l'Université de York en Angleterre.
- 2005 : GDR ISIS et GDR STIC-Santé : Co-organisateur de la journée : *Archivage et transmission sécurisés de l'information médicale*

5.2 Séminaires invités

- 1998 à l'Université de Toulon et du Var : *Reconstruction de surfaces courbes en vision monoculaire.*
- 2001 Institut Fresnel, Marseille : *Cryptage, Tatouage et Compression des images.*
- 2001 LIRMM, Robotique, Montpellier : *Tatouage des images pour la protection.*
- 2002 CUFRN, Nîmes : *Crypto-compression des images.*
- 2003 LIRMM, Informatique, Montpellier : *Cryptage des images et compression*
- 2004 : AS (Action Spécifique) Tatouage et contenu sécurisé, ENST Paris : *Protection des données par tatouage et cryptage.*
- 2004 LIRMM : *Système autonome par crypto-compression.*
- 2004 : GDR STIC-Santé, Hopital Georges Pompidou, Paris : *Compression et cryptage en imagerie médicale.*
- 2004 : GDR ISIS, thème compression, ENST Paris : *Codage conjoint d'images cryptage et compression.*
- 2005 : GDR ALP : Journées Codage et Cryptographie, Aussois : *Autonomie d'un chiffrement par flot asynchrone d'image par tatouage.*
- 2005 : Participation à une Ecole Jeunes Chercheurs en Informatique, Université Montpellier II, GDR ALP. Cours d'1 h sur le cryptage et les images.

Chapitre 6

Contrats et collaborations avec des entreprises

Durant ces dernières années de recherche, j'ai réussi à mettre en place des contrats avec des entreprises et des projets. Les sommes d'argent obtenues ont permis à mes étudiants et à moi-même de nous déplacer, principalement en Europe, afin de présenter nos résultats dans de nombreuses conférences.

- Financement Région par la région Provence, Alpes, Côte d'Azur pour une collaboration avec le CHI Fréjus Saint-Raphaël, 1999, **13.333 euros**.
- Prix Jeunes Chercheurs, UMII, 2001, **3.000 euros**.
- Collaboration Entreprise STRATEGIES, Paris : détection de peaux d'animaux pour la découpe de chaussures, 2003, **8.000 euros**.
- Action Spécifique du CNRS, Contenu sécurisé et Tatouage, 2004, **1.900 euros**.
- Collaboration Entreprise E-Desk, Montpellier : Sécurisation de documents électroniques par insertion d'une signature électronique dans une image de signature, 2004, **6.000 euros**, + partage des droits du logiciel (CNRS).
- Collaboration Entreprise STRATEGIES, Paris : Détection de peaux d'animaux pour la découpe de chaussures, suite, 2004, **8.000 euros**.
- Collaboration Entreprise SIGILLUM, Montpellier : Développement d'une méthode autonome de cryptage d'images, 2005, **10.000 euros**.
- Collaboration Entreprise STRATEGIES, Paris : Reconstruction et Sécurisation d'un objet 3D, 2005-2007, **30.000 euros**, contrat associé à une bourse Cifre.

Chapitre 7

Projets en cours

Dans cette partie, je décris les projets en cours de développement que je souhaite poursuivre durant les prochaines années. La plupart de ces projets sont d'ordre structurel et organisationnel.

7.1 Collaboration avec l'Entreprise STRATEGIES

Depuis 3 ans, je collabore avec la société STRATEGIES. Cette société parisienne de 25 personnes, est spécialisée en CAO pour l'industrie de la chaussure et pour la gestion de patrimoines. Notre collaboration en cours s'est concrétisée par la thèse de Philippe Amat (bourse cifre) qui développe un système de numérisation et sécurisation de modèles 3D de chaussures. La protection d'objets numériques a de nombreuses applications dans le monde industriel. La société STRATEGIES compte, avec notre collaboration, se spécialiser dans le domaine de la protection d'objets 3D afin de proposer sur le marché de nouveaux produits innovants en terme de sécurité.

7.2 Appels à Projet 2005

Cette année j'ai participé au montage d'un certain nombre de projets nationaux et d'un projet européen.

Projet ALTVis3D - RNTL. Ce projet, porté par des chercheurs du LSIS, UMR CNRS, Université de Marseille, consiste à développer une architecture logicielle pour la visualisation 3D temps réel en simulation technico-opérationnelle impliquant des données géoréférencées multi-sources hétérogènes. Dans cette application la visualisation en temps réel de modèles numériques de terrains avec enrichissement de données est difficile à obtenir. Il m'a été proposé de participer à ce projet dans le

but d'insérer des données cachées dans les cartes de terrains afin de synchroniser les altitudes avec la texture et une approche multirésolution. De plus pour des accès en ligne, la protection du transfert devra être assurée par cryptage partiel et rapide des données. Les entreprises SII (Aix) et PIXXIM (Marseille) sont associées à ce projet.

Projet TSAR - ARA SSIA. Ce projet, porté par des chercheurs de l'IRCCyN, UMR CNRS à Nantes, consiste à développer un système d'accès multirésolution à une base de données de peintures numériques. Les partenaires de ce projet sont le C2RMF, UMR CNRS (musée du Louvre), Paris, le LIS, UMR CNRS, Grenoble et l'ERIT à Rennes. J'ai participé fortement à la rédaction de ce projet l'an passé. Ce projet est l'occasion de travailler pour une application originale qui est la base de données EROS, base de données du musée du Louvre.

Projet VIDEOPROTECT - RIAM. Ce projet, porté par Henri Nicolas, du LABRI, UMR CNRS, Bordeaux, a pour objectif de développer des nouvelles approches de protection de vidéos. Ces approches combineront l'analyse de scènes, le tatouage et le cryptage partiel de vidéos. Le troisième partenaire de ce projet est Jean-Luc Dugelay de l'Institut EURECOM, Sophia-Antipolis.

Projet VIETE - RNTL. Par l'intermédiaire de Jean-Claude Bajard, Professeur 27° au LIRMM, j'ai été intégré dans un projet codage et cryptographie. L'objectif de ce projet est de développer des mathématiques de la transmission et des échanges de données. Les porteurs de ce projet sont l'IML (Institut de Mathématiques de Luminy), UMR CNRS, Marseille.

Projet européen - 3DART. André Crosnier, Professeur 61° au LIRMM m'a proposé d'intégrer un projet européen ayant pour objectif de créer des nouveaux systèmes d'acquisition, de visualisation et de gestion d'oeuvres d'art numériques.

7.3 Collaboration avec le Laboratoire Computer Vision

J'envisage également, dans le cas de l'obtention d'une délégation ou d'un détachement CNRS, d'aller effectuer un séjour à l'Université de York en Angleterre dans le laboratoire de Computer Vision and Pattern Recognition (CVPR), dirigé par le professeur Edwin Hancok. J'ai démarré depuis 3 ans des discussions avec un chercheur du CVPR, Adrian Bors. Adrian Bors et moi-même souhaitons développer des travaux de recherche en commun dans le domaine de l'insertion de données dans des objets 3D. Dans un premier temps, Adrian Bors m'a proposé d'accueillir un des doctorants que je co-encadre pour un séjour

de deux mois (juin et juillet 2005). J'ai été lui rendre visite à la fin de son séjour. Dans un deuxième temps j'espère pouvoir l'accueillir au LIRMM pour une durée d'un mois. Enfin, j'irai à l'Université de York pour continuer cet échange.

7.4 Direction du projet ICAR (Image et Réalité virtuelle) du LIRMM

Le laboratoire LIRMM a souhaité développer une équipe-projet de recherche dans le domaine de l'Image et de la Réalité virtuelle que je dirige depuis 2 ans. En effet, des chercheurs travaillent dans le domaine de l'image dans les trois départements du LIRMM (Informatique, Robotique et Microélectronique). Les thématiques Image, Signal, Vision, 3D, Informatique graphique et Réalité virtuelle ne sont pas visibles de l'extérieur. Cependant, depuis plus de 10 ans, 4 chercheurs environ travaillent dans ces domaines. Dès mon intégration au laboratoire, l'ancien directeur du LIRMM, Michel Habib, ainsi que le chef du département Robotique, Etienne Dombre, ont souhaité que je sois un élément actif dans cette opération. Le directeur actuel, Michel Robert, a fait mettre en place un conseil scientifique au niveau du laboratoire. Ce conseil propose d'aider la construction d'équipes-projets avec des thèmes transversaux. Le projet ICAR que je dirige comporte actuellement 9 personnes dont 1 Professeur 61, 1 CR 7, 2 MCF 27 et 5 MCF 61.

Dans les prochaine années, je souhaite consolider le projet ICAR en augmentant les échanges scientifiques des chercheurs de ce projet. Mon objectif est que les thématiques Image et Réalité virtuelle du LIRMM deviennent lisibles et visibles d'un point de vue national et international.

7.5 Rédaction d'un chapitre de livre

Dans le cadre de la série "Traitement du Signal et de l'Image" des traités IC2 (Information - Commande - Communication), publications Hermes, j'ai été sollicité pour rédiger un chapitre de livre concernant la compression des images médicales. Il m'a été proposé de prendre en charge un chapitre intitulé "protection des images médicales par crypto-compression". Ce livre doit être terminé en 2006 afin de paraître début 2007.

7.6 Co-organisation de la conférence CORESA

La conférence CORESA (COmpression et Représentation des Signaux Audiovisuels) est soutenue depuis 10 ans par France Télécom. Nous avons en projet d'accueillir et d'organiser

cette conférence à Montpellier au LIRMM en 2007. En effet, suite à un accord avec les responsables de France Télécom, j'ai été nommé responsable de l'organisation de cette conférence. Ce colloque regroupe entre 100 et 200 personnes sur 2 jours.

7.7 Participation à la création du Master Réseaux et Images

Actuellement, un parcours de Master intitulé "Réseaux et Images" se construit entre Montpellier et Nîmes. Je fais partie du groupe de travail qui a pour mission de proposer un contenu pédagogique pour ce parcours. Depuis 3 ans je travaille dans ce sens et j'espère voir ouvrir assez rapidement ce parcours sur les sites de Nîmes et Montpellier.

Chapitre 8

Résumé de mes activités de recherche

Introduction

Dans ce chapitre je résume le contenu de la seconde partie de mon document concernant mes activités de recherche[1]. La partie II est composée de 5 chapitres.

La mise en place d'interface de visualisation à distance de données connaît une forte demande depuis ces 10 dernières années. Ces interfaces permettent en général d'accéder à des dossiers contenant des informations textuelles, graphiques et sonores. Le développement de ce type de système soulève un nombre conséquent de problèmes qui ne sont pas tous encore résolus. Un premier problème concerne le temps de transfert. La qualité des données transmises dépend fortement du temps de transfert alloué pour l'application. En effet pour des raisons de temps de transfert au travers des réseaux toutes les données, et en particulier les images, doivent être comprimées. En fonction des applications la compression pourra être plus ou moins importante. Par exemple, dans le cas d'application de traitement de données en temps réel, la compression de données importante sera inévitable. Un deuxième problème concerne l'aspect sécurité pendant le transfert des données, mais également après réception de celles-ci : il ne faut absolument pas que des données puissent être dissociées les unes des autres afin d'éviter toute confusion durant la phase de réception. De plus, pour des raisons de confidentialité, ces données doivent être rendues complètement ou partiellement illisibles et non déchiffrables pendant le transfert.

Ma mobilité géographique a eu des répercussions sur mes thématiques scientifiques. J'illustre figure 1 l'évolution chronologique des mes thématiques entre 1997 et 2005. Dans un premier temps, entre 1997 et 2000, associé à des médecins j'ai développé des systèmes

1. Dans ce chapitre, j'ai inséré les introductions de 5 chapitres de la partie II.

de traitement à distance des images en haute résolution, figure 1.a. Dans un second temps, toujours en 1997 et 2000, j'ai pris en compte l'aspect visualisation 3D à distance en particulier pour des organes anatomiques à partir de séquences d'images scanner, figure 1.b. Dès mon arrivée sur Nîmes en 2000, l'aspect sécurité des transferts d'images a été primordial. Je me suis donc orienté vers des approches de marquage et de chiffrement des images, figure 1.c. Enfin, depuis plus de 2 ans, intégré au LIRMM, je privilégie le développement de méthodes permettant le transfert rapide et sécurisé d'images pour des environnements de faible puissance comme illustré figure 1.d.

Dans le chapitre 1 je présente mes travaux de traitements d'images médicales à distance qui ont été développés en collaboration avec le CHI de Fréjus-Saint Raphaël. Dans le chapitre 2 je développe des méthodes d'analyse et de reconstruction 3D. Je présente, chapitre 3, des méthodes de protection de données par insertion de données cachées. Le chapitre 4 est consacré aux algorithmes de cryptage appliqués aux images. Enfin, je présente chapitre 5 des méthodes de codage hybride combinant cryptage, insertion de données cachées et compression.

8.1 Traitements d'images médicales à distance pour l'aide aux télédiagnostics

Dans ce premier chapitre je présente les travaux effectués à l'Université de Toulon et du Var concernant du traitement d'images à distance. L'application concernait la mise en place d'un réseau d'images médicales développé en collaboration avec le CHI (Centre Hospitalier Intercommunal) Fréjus Saint Raphaël et le CHI Toulon la Seyne-sur-mer. L'objectif de ces recherches était d'apporter une aide au télé-diagnostic. Dans ce chapitre je développe la partie réseau concernant le traitement d'images à distance. Ensuite, après avoir analysé les méthodes existantes dans les centres hospitaliers, nous présentons des nouvelles méthodes de détection de contours en vue d'une reconstruction 3D automatique d'un organe anatomique.

Nous décrivons le réseau mis en place au CHI de Fréjus - St Raphaël pour la mise à disposition et la visualisation d'images médicales numériques issues de divers appareils de radiologie. L'aspect novateur réside dans l'adaptation de logiciels du domaine public à un parc d'ordinateurs PC basiques. Ce réseau unique en France (en 1998) permet, par une baisse sensible des coûts, d'envisager sa banalisation.

Ces travaux ont été développés avec **S. Nicolay** et **J. Michelis** dans le cadre de leur stage de DEA. Cette partie a donné lieu aux publications suivantes : [Ricordel 99, Nicolay 99, Puech 99, Puech 00, Michelis 00, Bouchouicha 00].

8.2 Détection de contours et reconstruction 3D

Dans ce chapitre, je présente mes travaux sur la détection et le suivi de contours d'objets déformables, pour l'étude d'une séquence de coupes issues d'un appareil tomodensitomètre à rayons X. Dans un premier temps, nous avons analysé les méthodes utilisées dans les services d'imagerie médicale. Elles sont principalement basées sur des techniques de seuillage et de soustraction d'images. Nous avons d'abord proposé d'améliorer cette méthode en réalisant semi-automatiquement l'extraction d'une seule structure anatomique ciblée. Notre méthode vise à mettre en oeuvre un modèle de contours actifs et procède en trois étapes : la détection d'un contour sur la première coupe, la propagation de ce contour en le déformant aux coupes connexes et la reconstruction 3D.

Dans ce chapitre, nous présentons également un nouvel algorithme de suivi d'un organe dans une séquence d'images médicales afin de réaliser une reconstruction 3D. La méthode automatique que nous proposons permet de suivre le contour externe d'un organe anatomique dans toute la séquence à partir d'un contour initialisé par l'utilisateur sur la première image. Les opérations nécessaires pour notre méthode de suivi s'appuient sur une segmentation par contours actifs basée région. La localisation des objets avec une prédiction dynamique de déplacement est basée sur les fonctions de courbes de niveaux et sur la définition de région d'intérêt pour l'estimation locale robuste du modèle de l'image. Une application de cette méthode est la reconstruction 3D de l'aorte abdominale.

Dans la section 2.1 nous présentons une analyse et des améliorations de méthodes de reconstruction 3D de l'aorte à partir d'une séquence d'images tomodensitométriques. Nous détaillerons, section 2.2, une méthode permettant de propager automatiquement des contours actifs dans une séquence d'images médicales.

Ces travaux ont été développés avec **G. Ledanff** dans le cadre de son stage de DEA et avec **K. Djemal** dans le cadre de sa thèse. Cette partie a donné lieu aux publications suivantes : [Nicolay 99, Puech 99, Puech 00, Djemal 02, Djemal 03b, Djemal 03a, Djemal 04, Djemal 05].

8.3 Protection des données par insertion de données cachées dans des images

La mise en place d'interfaces de visualisation à distance connaît actuellement une forte demande dans le cas de transfert de données textuelles et images. Le premier problème rencontré concerne la qualité des données transmises. En effet, pour des raisons de temps de transfert au travers du réseau, toutes les données, et en particulier les images, sont comprimées. Le deuxième problème concerne l'aspect sécurité. La sécurisation des images devient extrêmement importante pour de nombreuses applications comme les transmissions confidentielles, la vidéo surveillance et les applications militaires et médicales. Par exemple, la nécessité d'un diagnostic rapide et sûr est vital dans le monde médical [Bernarding 01, Norcen 03]. Pendant le transfert, dans certaines applications, il ne faut absolument pas qu'une image soit dissociée des informations textuelles. De plus, pour des raisons de confidentialité, ces données doivent être rendues illisibles et non déchiffrables, donc cryptées. Dans ce chapitre, je présente des nouvelles méthodes d'insertion de données cachées (IDC) robustes à la compression. Dans le cadre de la protection de données la capacité d'insertion est relativement importante et peut atteindre 10% de la taille de l'image support. Afin d'être robuste à la compression, nous nous sommes orientés vers des approches basées sur la DCT (Discrete Cosinus Transform). Dans le domaine fréquentiel les possibilités d'insertion de données sont nombreuses mais dépendent principalement du lieu d'insertion pouvant varier de la composante continue jusqu'aux très hautes fréquences. Concernant le choix des fréquences, nous avons opté pour une insertion au niveau des basses fréquences ou de la composante continue. De ce fait, la marque est plus robuste aux diverses transformations que l'image peut subir (compression plus importante, lissage, bruit et rehaussement de contraste). Par contre, le fait d'insérer les données dans les basses fréquences dégrade plus l'image. Pour cela, nos travaux de recherche ont consisté à trouver des méthodes permettant de dégrader le moins possible la qualité de l'image. Dans cette partie nous proposons donc une méthode inductive d'IDC combinant le domaine fréquentiel avec le domaine spatial. Nous montrons que la qualité de l'image est meilleure que dans le cas d'une approche classique d'IDC. Afin de résister à des attaques désynchronisantes (translation, rotation, découpage et changement d'échelle), je propose également, dans ce chapitre, une approche d'insertion de données cachées basée sur le contenu. Cette approche permet d'insérer des données particulières dans chaque région d'intérêt contenue dans l'image.

Dans la section 3.2, je présente les grandes classes d'IDC. Dans la section 3.3, j'approfondis des méthodes d'IDC basées sur la DCT. Section 3.4, je développe une nouvelle méthode d'IDC combinée avec JPEG permettant d'améliorer la qualité des images marquées et comprimées par rapport aux méthodes classiques. Dans la section 3.5, je présente une analyse quantitative théorique et expérimentale de l'amélioration de la qualité des images marquées par la méthode proposée. Dans la section 3.6, j'étends cette méthode aux images couleurs et je propose de m'appuyer sur le contenu des images pour effectuer l'IDC.

Ces travaux ont été développés avec **G. Lo-Varco** dans le cadre de son stage de DEA et de sa thèse ainsi qu'avec **JL. Toutant** dans le cadre de son stage de DEA et de sa thèse et **Ph. Amat** dans le cadre de son stage de DEA. Cette partie a donné lieu aux publications suivantes : [Puech 01c, Puech 01d, Puech 02, Lovarco 03a, Lovarco 03b, Puech 03, Lovarco 04b, Rodrigues 04b, Lovarco 04a, Lovarco 05c, Lovarco 05a, Lovarco 05b, Toutant 05a, Toutant 05b, Amat 05].

8.4 Cryptage d'images

Dans ce chapitre, nous montrons comment les algorithmes classiques de chiffrement peuvent être appliqués à des images. Les données images sont des données particulières du fait de la taille des images et de l'information bidimensionnelle. Nous présentons de nombreux algorithmes par bloc ou par flot symétrique ou asymétrique. Nous concluons que les algorithmes asymétriques tel que le RSA ne sont pas adaptés aux images du fait de leur complexité dû à l'utilisation de grands nombres premiers car une partie de la clef est connue (clef publique). Concernant les algorithmes symétriques, les méthodes par bloc présentent des inconvénients quand l'image contient des zones homogènes. Dans le cas des algorithmes de chiffrement par flot, les zones homogènes ne sont plus visibles dans l'image cryptée. De plus les chiffrements par flot sont très rapides. Cependant, quelque soit l'algorithme de cryptage utilisé, il est alors difficile de comprimer l'image puisque théoriquement les redondances ont été supprimées durant la phase de cryptage et donc l'entropie devient maximale. De plus les algorithmes de chiffrement par bloc supportent très mal le bruit, en effet dès qu'un bit d'un bloc est altéré alors le bloc complet n'est pas décryptable. Dans le cas des chiffrements par flot, la robustesse au bruit semble plus importante. Dans ce chapitre nous présentons également une première approche de crypto-compression basée sur des images contenant des zones homogènes. Le premier objectif de

cette méthode était de faire disparaître les zones homogènes, mais au final l'image est comprimée sans perte.

Les analyses développées dans ce chapitre sont à la base des méthodes de codage conjoint que nous présentons dans le chapitre suivant. L'objectif est alors de combiner les processus de compression et de cryptage.

Ces travaux ont été développés avec **S. Piat** et **G. Benoît** dans le cadre de leur stage de DEA ainsi qu'avec **JC. Borie** dans le cadre de sa thèse.

Cette partie a donné lieu aux publications suivantes : [Puech 01c, Puech 01b, Puech 01a, Puech 01d, Borie 02a, Borie 02b, Borie 04b, Borie 04a].

8.5 Codage hybride cryptage-insertion de données cachées et compression

Dans ce chapitre je développe des nouvelles méthodes de codage originales combinant toutes au moins deux types de codage différents, à savoir cryptage, insertion de données cachées et compression. Ces méthodes ont toutes pour objectif de protéger des données et sont issues des travaux présentés dans les chapitres 3 et 4.

La première méthode proposée, section 5.2, combine cryptage d'images et insertion de données cachées afin de rendre autonome un système de transmission sécurisé d'images. En effet, dans une approche classique à clef secrète, il faut utiliser un autre canal de transmission pour transférer la clef. A partir d'un algorithme de chiffrement par flot asynchrone robuste au bruit, nous proposons d'insérer dans l'image cryptée la clef secrète chiffrée par un algorithme asymétrique. Nous avons rappelé que les méthodes asymétriques ne conviennent pas aux images car trop longues en temps de calcul.

La seconde méthode, section 5.3, propose de combiner cryptage, compression et insertion de données cachées en créant un nouveau format d'image. Nous montrons dans cette méthode qu'en découpant l'image en deux parties (4 plans binaires de poids fort et 4 plans binaires de poids faible) il était possible dans la partie haute (plans binaires de poids forts) de l'image d'effectuer à la fois de l'insertion de données cachées et de la compression.

La troisième méthode, présentée section 5.4, propose de protéger la haute résolution d'une région d'intérêt de l'image fortement comprimée par JPEG. Actuellement, il est possible avec JPEG2000 de ne pas comprimer une région d'intérêt de l'image tout en comprimant fortement le reste de l'image. Dans ce cas, à la décompression toute l'information

est visible. Dans le cas de notre approche, la haute résolution n'est visible que si la personne qui décomprime l'image possède la clef secrète. En effet, nous évaluons la quantité de données perdues dues à la compression et nous insérons par données cachées ces pertes dans l'image comprimée.

Enfin, dans ce chapitre je propose, section 5.5, une méthode permettant de crypter de manière sélective les données de l'image tout en conservant un niveau de sécurité suffisant. Les avantages sont de pouvoir garder le taux de compression initial de l'image et de gagner en temps de calcul. En effet dans notre approche le cryptage des données est réellement effectué en même temps que la compression et ne rajoute aucune donnée supplémentaire.

Ces travaux ont été développés avec **S. Martineau**, **O. Léger**, **D. Falguère** et **A. Martin** dans le cadre de leur stage de DEA ainsi qu'avec **J. Rodrigues** dans le cadre de sa thèse. Cette partie a donné lieu aux publications suivantes : [Rodrigues 06, Puech 06, Amat 05, Toutant 05a, Puech 05, Puech 04a, Rodrigues 04a, Puech 04b].

Chapitre 9

Liste des publications

9.1 Brevet

Brevet déposé par le CNRS en collaboration avec la société SIGILLUM Technologies, (Septembre 2005) 123-01. Transfert sécurisé et autonome d'images.

9.2 Chapitre de livre

W. Puech. Compression des images et des signaux médicaux. Chapitre 9 : Sécurisation des données, Edition Hermès, Traité IC2, en cours de rédaction, 2007.

9.3 Revues internationales avec comité de lecture

Rodrigues 06 RI J.M. Rodrigues and W. Puech. An Adaptable Invertible Crypto-Data Hiding Method for Still Heterogeneous Images. EURASIP Journal on Applied Signal Processing, in revision.

Djemal 06 RI K. Djemal, W. Puech and B. Rossetto. Automatic Active Contours Propagation in a Sequence of Medical Images. International Journal of Image and Graphics (IJIG), vol.6 , n° 1, 2006, january 2006.

Lo-varco 05 RI G. Lo-varco, W. Puech and M. Dumas. Content Based Watermarking for Securing Color Images. Journal of Imaging Science and Technology (JIST), vol. 49 , n° 5, pp. 450-459, september 2005.

Puech 01 RI W. Puech, A. Bors, I. Pitas and J.-M. Chassery. Projection Distortion Analysis for Flattened Image Mosaicing from Straight Uniform Generalized Cylinders. Pattern Recognition (PR), vol. 34, n° 8, pp. 1657-1670, august 2001.

Puech 97 PO W. Puech, J.-M. Chassery and I. Pitas. Curved Surface Reconstruction

Based on Parallels. A. Le Mehaute, C. Rabut and L.L. Schumaker Curves and Surfaces in Geometric Design, Vanderbilt University Press, Nashville, TN, pp. 481-488, 1997.

Puech 97 RI W. Puech, J.-M. Chassery and I. Pitas. Cylindrical Surface Localization in Monocular Vision. Pattern Recognition Letters (PRL), vol. 18, n° 8, pp.711-722, august 1997.

Soumises

Puech 06 RI W. Puech and J.M. Rodrigues. An Autonomous Crypto-Data Hiding Method for Images Safe Transfer. Signal Processing: Image Communication (SPIC), submitted.

Rodrigues 06 RI J.M. Rodrigues and W. Puech. A Scalable Selective Crypto-Compression Based in JPEG Huffman Coding and AES Cipher in OFB Mode. Journal of Visual Communication and Image Representation (JVCI), submitted.

9.4 Revues nationales avec comité de lecture

Soumises

Puech 06 RN W. Puech and J.M. Rodrigues. Transfert sécurisé d'images médicales par codage conjoint : cryptage sélectif par AES en mode par flot et compression JPEG. Traitement du signal (TS), numéro spécial "Traitement du signal appliqué à la cancérologie", submitted.

9.5 Conférences internationales avec actes et comité de lecture

Lo-Varco 05 CI b G. Lo-varco and W. Puech. DCT-Based Data-Hiding for Securing ROI of Color Images. *Proc. International Conference on Image Processing* **IEEE ICIP-2005**, pp. 1086-1089, Genova, Italy, september 2005.

Puech 05 CI W. Puech and J.M. Rodrigues. Crypto-Compression of Medical Images by Selective Encryption of DCT. *13th European Signal Processing Conference,* **EUSIPCO'05**, Antalya, Turkey, september 2005.

Lo-Varco 05 CI a G. Lo-Varco and W. Puech. Safe ROIs of Color Images by Inductive Data-Hiding. *13th European Signal Processing Conference,* **EUSIPCO'05**, Antalya, Turkey, september 2005.

Joffre 05 CI J. Joffre, W. Puech, F. Comby and J. Joffre. High Dynamic Range Images from Digital Cameras Raw Data. *Proc. 32th International Conference on Computer Graphics and Interactive Techniques* **SIGGRAPH 2005**, Los Angeles, USA, july 2005.

Chouchane 05 CI S. Chouchane and W. Puech. A Multi-Watermarking Method Based on Wavelets Combined with the EZW Coder. *Proc. IEEE International Computer Systems and Information Technology Conference, ICSIT'05*, Algiers, Algeria, july 2005.

Toutant 05 CI J.L. Toutant, W. Puech and C. Fiorio. Asynchronous DCT-Based Data-Hiding Robust to Cropping. *Proc. 6th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'05*, Montreux, Switzerland, april 2005.

Borie 04 CI b J.C. Borie, W. Puech and M. Dumas. Crypto-Compression Using TEA's Algorithm and a RLC Compression. *Proc. 2nd Intelligent Access to the Multimedia Documents on the Internet, MediaNet'04*, pp. 5-16, Tozeur, Tunisia, november 2004.

Puech 04 CI W. Puech and J.M. Rodrigues. A New Crypto-Watermarking Method for Medical Images Safe Transfer. *Proc. 12th European Signal Processing Conference,* **EUSIPCO'04**, pp. 1481-1484, Vienna, Austria, september 2004.

Borie 04 CI a J.C. Borie, W. Puech and M. Dumas. Crypto-Compression System for Secure Transfer of Medical Images. *Proc. 2nd International Conference on Advances in Medical Signal and Information Processing, MEDSIP'04*, pp. 327-331, Malte, september 2004.

Rodrigues 04 CI b J.M. Rodrigues, W. Puech and C. Fiorio. Lossless Crypto-Data Hiding in Medical Images Without Increasing the Original Size. *Proc. 2nd International Conference on Advances in Medical Signal and Information Processing, MEDSIP'04*, pp. 358-365, Malte, september 2004.

Lo-varco 04 CI G. Lo-Varco, W. Puech and M. Dumas. DCT-Based Watermarking Method Using Color Components. *Proc. 2nd European Conference on Colour in Graphics, Imaging and Vision,* **CGIV'04**, pp. 146-150, Aachen, Germany, april 2004.

Rodrigues 04 CI a J.M. Rodrigues, J.R. Rios and W. Puech. SSB-4 System of Steganography using Bit 4. *Proc. 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'04*, Lisboa, Portugal, april 2004.

Djemal 04 CI K. Djemal, W. Puech and B. Rossetto. Geometric Active Contour Model Using Level Set Methods for Objects Tracking in Images Sequences. *Proc. 2nd*

International Conference of Sciences of Electronic, Technology of Informations and Telecommunications, SETIT'04, Sousse, Tunisia, march 2004.

Lo-Varco 03 CI G. Lo-Varco, W. Puech and M. Dumas. DCT-Based Watermarking Method using Error Correction Coding. *Proc. 5th International Conference on Advances in Pattern Recognition,* **ICAPR'03**, pp. 347-350, Calcutta, India, december 2003.

Bouchouicha 03 CI M. Bouchouicha, M. Ben Khelifa and W. Puech. A Non-Linear Camera Calibration With Genetic Algorithms. *Proc. 7th International Symposium on Signal Processing and its Applications, ISSPA'03*, Paris, France, july 2003.

Puech 03 CI W. Puech. Safe Transfer of Image Based on Color Transformation for Watermarking. *Proc. Workshop Transmitting, Processing and Watermarking Multimedia Content, WTPWMC'03*, pp. 1-6, Bordeaux, France, march 2003.

Djemal 03 CI K. Djemal, S. Paris, M. Grimaldi, W. Puech and B. Rossetto. Réseau d'imagerie médicale et d'aide au diagnostic (RIMAD). *Proc. 1st International Conference of Sciences of Electronic, Technology of Informations and Telecommunications, SETIT'03*, Sousse, Tunisie, march 2003.

Borie 02 CI b J.C. Borie, W. Puech and M. Dumas. Encrypted Images for Secure Transfer with RSA Algorithm. **Proc. IEEE Communication**, Bucharest, Romania, december 2002.

Djemal 02 CI K. Djemal, W. Puech and B. Rossetto. Active Contours Propagation in a Medical Images Sequence with a Local Estimation. *Proc. 11th European Signal Processing Conference,* **EUSIPCO'02**, Toulouse, France, 03-06 september 2002.

Borie 02 CI a J.C. Borie, W. Puech and M. Dumas. Encrypted medical images for secure transfer. *Proc. International Conference Diagnostic Imaging and Analysis, ICDIA'02*, Shanghai, P.R. China, august 2002.

Puech 02 CI W. Puech, P. Montesinos and M. Dumas. Color Image Watermarking Robust to JPEG Compression. *Proc. 1st European Conference on Colour in Graphics, Imaging and Vision,* **CGIV'02**, Poitiers, France, april 2002.

Puech 00 CI W. Puech G. Passail and V. Ricordel. Analysis and Optimisation of 3D Reconstruction Method of the Aorta from a Tomographic Images Sequence.*Proc. 10th European Signal Processing Conference,* **EUSIPCO'2000**, vol. 1, Tampere, Finlande, september 2000.

Puech 98 CI W. Puech and J.-M. Chassery. Curvature Variation of Projected Cross-Sections From Straight Uniform Generalized Cylinders. *Proc. 9th European Signal*

Processing Conference, **EUSIPCO'98**, vol.4, pp. 2181-2184, Rhodes, Grèce, september 1998.

Bors 97 CI A. G. Bors, W. Puech, I. Pitas and J.-M. Chassery. Mosaicing of Flattened Images from Curved Surfaces. *Proc. 7th International Conference Computer Analysis of Images and Patterns, CAIP'97*, pp. 122-129, Kiel, Germany, september 1997.

Puech 97 CI W. Puech, and J.-M. Chassery. A New curved Surface Localization Method Using a Single Perspective View. *Proc. 10th candinavian Conference on Image Analysis, SCIA'97*, pp. 44-49, Lappeenranta, Finland, june 1997.

Bors 97 CI A. G. Bors, W. Puech, I. Pitas and J.-M. Chassery. Perspective Distortion Analysis for Mosaicing Images from Cylindrical Surfaces. *Proc. 3rd IEEE International Conference on Acoustics, Speech, and Signal Processing,* **ICASSP 97**, pp. 3049-3052, Munich, Germany, april 1997.

Puech 96 CI W. Puech, A. G. Bors, I. Pitas and J.-M. Chassery. Mosaicing of Painting on Curved Surfaces. *Proc. 3rd IEEE Workshop on Applications of Computer Vision, WACV'96*, pp. 44-49, Sarasota, Florida, USA, december 1996.

Puech 96 CI W. Puech and J.-M. Chassery. Curved Surface Reconstruction Using Monocular Vision. *Proc. 8th European Signal Processing Conference,* **EUSIPCO'96**, vol. 1, pp. 9-12, Trieste, Italy, september 1996.

Puech 96 CI W. Puech, J.-M. Chassery and I. Pitas. Curved Surface Reconstruction Based on Parallels. *Proc. 3rd International Conference on Curves and Surfaces*, pp. 48, Chamonix, France, june 1996.

9.6 Conférences nationales avec actes et comité de lecture

Chouchane 05 CN S. Chouchane and W. Puech. Intégration d'un nouveau marqueur dans le codeur d'images EZW basé sur les ondelettes. *Proc. 10th Colloque Compression et Représentation des Signaux Audiovisuels,* **CORESA'05**, Rennes, France, november 2005.

Amat 05 CN P. Amat and W. Puech. Transfert sécurisé d'une RI sans perte par une méthode d'insertion de données cachées robuste à la compression JPEG. *Proc. 20th. Colloque Traitement du Signal et des Images* **GRETSI'05**, pp. 1045-1048, Louvain-la-Neuve, Belgique, september 2005.

Toutant 05 CN J.L. Toutant, W. Puech and C. Fiorio. Amélioration de l'invisibilité

par adaptation de la quantification aux données à insérer. *Proc. 20th. Colloque Traitement du Signal et des Images* **GRETSI'05**, pp. 1193-1196, Louvain-la-Neuve, Belgique, september 2005.

Puech 05 CN W. Puech and J.M. Rodrigues. Crypto-compression d'images médicales par cryptage partiel des coefficients DCT. *Journées Sciences, Technologies et Imagerie pour la Médecine, JSTIM*, pp 149-150. Nancy, France, march 2005.

Lovarco 04 CN G. Lovarco, W. Puech and M. Dumas. Tatouage couleur par DCT basé sur le contenu. *Proc. 9th Colloque Compression et Représentation des Signaux Audiovisuels,* **CORESA'04**, pp. 13-16, Lille, France, may 2004.

Puech 04 CN W. Puech and J.M. Rodrigues. Sécurisation d'image par crypto-tatouage. *Proc. 9th Colloque Compression et Représentation des Signaux Audiovisuels,* **CORESA'04**, pp. 215-218, Lille, France, may 2004.

Djemal 03 CN K. Djemal, W. Puech and B. Rossetto. Restauration par minimisation de la variation totale adaptée à un modèle de bruit ultrasonore. *Proc. 19th. Colloque Traitement du Signal et des Images* **GRETSI'03**, Paris, France, september 2003.

Lovarco 03 CN G. Lovarco, W. Puech and M. Dumas. Tatouage d'images couleurs avec CCE : application à la sécurité routière. *Proc. 8th Colloque Compression et Représentation des Signaux Audiovisuels,* **CORESA'03**, Lyon, France, january 2003.

Puech 01 CN d W. Puech, J.J. Charre and M. Dumas. Transfert sécurisé d'images par chiffrement de Vigenère. *Proc. Colloque Nimestic'01*, Nîmes, France, december 2001.

Puech 01 CN c W. Puech and M. Dumas. Transfert sécurisé d'images par combinaison de techniques de cryptographie et de tatouage. *Proc. 7th Colloque Compression et Représentation des Signaux Audiovisuels,* **CORESA'01**, Dijon, France, november 2001.

Puech 01 CN b W. Puech, M. Dumas, J.C.Borie and M. Puech. Tatouage d'images cryptées pour l'aide au Télédiagnostic. *Proc. 18th. Colloque Traitement du Signal et des Images* **GRETSI'01**, Toulouse, France, september 2001.

Puech 01 CN a W. Puech, M. Puech and M. Dumas. Accès sécurisé à distance d'images médicales haute résolution. *Proc. 11th. Forum des Jeunes Chercheurs en Génie Biologique et Médical*, pp. 72-73, Compiègne, France, june 2001.

Michelis 00 CN J. Michelis, W. Puech, V. Ricordel, G. Passail and M. Dumas. Intégration d'applet JAVA dans un réseau d'images médicales : aide au télédiagnostic. *Proc.*

6th Colloque Compression et Représentation des Signaux Audiovisuels, **CORESA'2000**, pp. 403-410, Poitiers, France, october 2000.

Bouchouicha 00 CN M. Bouchouicha, W. Puech, A. Kolesnikov, G. Passail and M. Dumas. Visualisation d'images haute résolution au travers d'un arpenteur : application à l'imagerie médicale. *Proc. 6th Colloque Compression et Représentation des Signaux Audiovisuels,* **CORESA'2000**, pp. 395-402, Poitiers, France, october 2000.

Puech 99 CN W. Puech, G. Passail, S. Nicolay and V. Ricordel. Analyse et amélioration de méthodes de reconstruction 3D de l'aorte à partir d'une séquence d'images tomodensitométriques. *Proc. 17th. Colloque Traitement du Signal et des Images* **GRETSI'99**, vol. 4, pp. 1053-1056, Vannes, France, september 1999.

Passail 99 CN V. Ricordel, A. Kolesnikov, G. Passail and W. Puech. Conception d'un réseau pour la communication d'images médicales. *Proc. 5th Colloque Compression et Représentation des Signaux Audiovisuels,* **CORESA'99**, pp. 229-236, Nice-Sophia, France, june 1999.

Nicolay 99 CN S. Nicolay, W. Puech, V. Ricordel and G. Passail. Reconstruction 3D de l'aorte issue d'une séquence d'images tomodensitométriques : analyse et amélioration par contours actifs. *Proc. Colloque National de Recherche en IUT, CNRIUT'99*, pp. 117-129, Aix en Provence, France, june 1999.

Puech 97 CN W. Puech, J.-M. Chassery, A. G. Bors and I. Pitas. Mosaïque de Peintures sur Surfaces Courbes. *Proc. 16th. Colloque Traitement du Signal et des Images* **GRETSI'97**, pp. 427-430, Grenoble, France, september 1997.

Puech 95 CN W. Puech, P. Schott and J.-M. Chassery. Discrete Method of Calculus Using Common Normals for Curved Surfaces Reconstruction. *Proc. 5th Discrete Geometry for Computer Imagery, DGCI'95*, pp. 82-92, Clermont-Ferrand, France, september 1995.

Puech 95 CN W. Puech and J.-M. Chassery. Détection d'Axe sur Surfaces en Vision Monoculaire. *Proc. 15th. Colloque Traitement du Signal et des Images* **GRETSI'95**, vol. 2, pp. 877-880, Juan Les Pins, France, september 1995.

Conclusion

Dans cette partie j'ai présenté mes activités de ces dix dernières années. Bien qu'ayant eu une mobilité géographique importante, j'ai essayé de garder une continuité thématique dans mes recherches et mes enseignements. Entouré d'une vingtaine d'étudiants durant ces dix années, j'ai toujours tenu à aller présenter au maximum nos travaux dans des colloques nationaux ou internationaux. La participation à ces colloques m'a permis d'entretenir des contacts au sein de la communauté scientifique Image. Depuis mon intégration au LIRMM, les relations avec des entreprises françaises sont devenues plus importantes. Celles-ci me permettent d'assurer un transfert de technologies et d'obtenir une aide financière pour continuer mes recherches. Je compte beaucoup sur la mise en place de l'équipe-projet ICAR du LIRMM pour continuer à développer mes travaux. Dans la partie suivante, je vais détailler mes activités scientifiques.

Deuxième partie

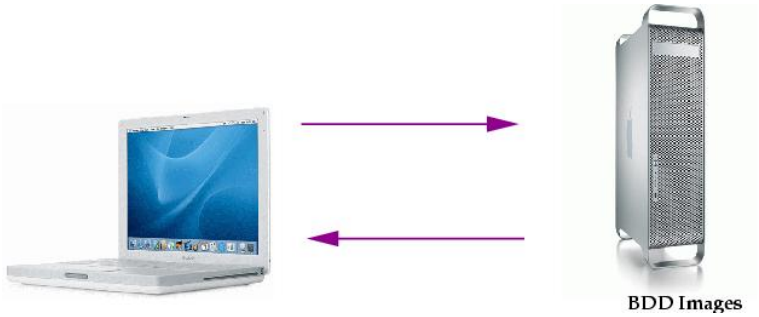
Activités de recherche

Introduction

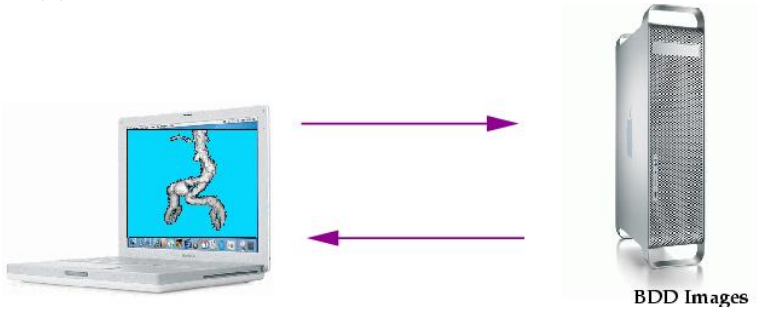
La mise en place d'interface de visualisation à distance de données connaît une forte demande depuis ces 10 dernières années. Ces interfaces permettent en général d'accéder à des dossiers contenant des informations textuelles, graphiques et sonores. Le développement de ce type de système soulève un nombre conséquent de problèmes qui ne sont pas tous encore résolus. Un premier problème concerne le temps de transfert. La qualité des données transmises dépend fortement du temps de transfert alloué pour l'application. En effet pour des raisons de temps de transfert au travers des réseaux toutes les données, et en particulier les images, doivent être comprimées. En fonction des applications la compression pourra être plus ou moins importante. Par exemple, dans le cas d'application de traitement de données en temps réel, la compression de données importante sera inévitable. Un deuxième problème concerne l'aspect sécurité pendant le transfert des données, mais également après réception de celles-ci : il ne faut absolument pas que des données puissent être dissociées les unes des autres afin d'éviter toute confusion durant la phase de réception. De plus, pour des raisons de confidentialité, ces données doivent être rendues complètement ou partiellement illisibles et non déchiffrables pendant le transfert.

Ma mobilité géographique a eu des répercussions sur mes thématiques scientifiques. J'illustre figure 1 l'évolution chronologique des mes thématiques entre 1997 et 2005. Dans un premier temps, entre 1997 et 2000, associé à des médecins j'ai développé des systèmes de traitement à distance des images en haute résolution, figure 1.a. Dans un second temps, toujours en 1997 et 2000, j'ai pris en compte l'aspect visualisation 3D à distance en particulier pour des organes anatomiques à partir de séquences d'images scanner, figure 1.b. Dès mon arrivée sur Nîmes en 2000, l'aspect sécurité des transferts d'images a été primordial. Je me suis donc orienté vers des approches de marquage et de chiffrement des images, figure 1.c. Enfin, depuis plus de 2 ans, intégré au LIRMM, je privilégie le développement de méthodes permettant le transfert rapide et sécurisé d'images pour des environnements de faible puissance comme illustré figure 1.d.

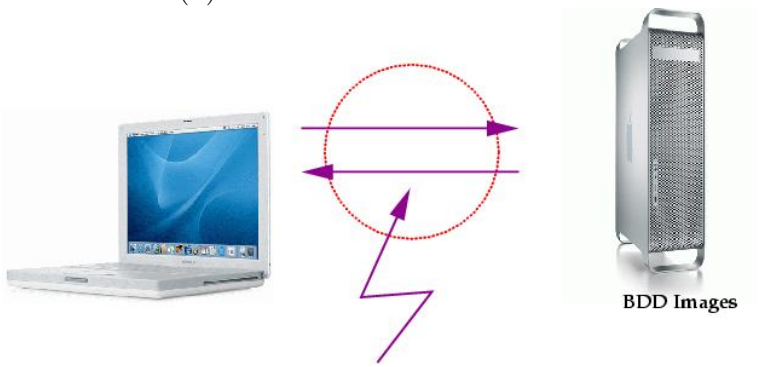
Cette partie est composée de 5 chapitres présentant en détail mes activités de recherche. Dans le chapitre 1 je présente mes travaux de traitements d'images médicales à distance qui ont été développés en collaboration avec le CHI de Fréjus-Saint Raphaël. Dans le chapitre 2 je développe des méthodes d'analyse et de reconstruction 3D. Je présente, chapitre 3, des méthodes de protection de données par insertion de données cachées. Le chapitre 4 est consacré aux algorithmes de cryptage appliqués aux images. Enfin, je présente chapitre 5 des méthodes de codage hybride combinant cryptage, insertion de données cachées et compression.



(a) Traitement à distance des images haute résolution.



(b) Visualisation 3D à distance.



(c) Traitement sécurisé des images par chiffrement et transfert sécurisé des données associées aux images.

(d) Traitement sécurisé à distance des images en temps réel pour des environnements faible puissance.

FIG. 1 – *Chronologie de mes thématiques scientifiques de 1997 à 2005.*

Chapitre 1

Traitements d'images médicales à distance pour l'aide aux télédiagnostics

Dans ce premier chapitre je présente les travaux effectués à l'Université de Toulon et du Var concernant du traitement d'images à distance. L'application concernait la mise en place d'un réseau d'images médicales développé en collaboration avec le CHI (Centre Hospitalier Intercommunal) Fréjus Saint Raphaël et le CHI Toulon la Seyne-sur-mer. L'objectif de ces recherches était d'apporter une aide au télé-diagnostic. Dans ce chapitre je développe la partie réseau concernant le traitement d'images à distance. Ensuite, après avoir analysé les méthodes existantes dans les centres hospitaliers, nous présentons des nouvelles méthodes de détection de contours en vue d'une reconstruction 3D automatique d'un organe anatomique.
Nous décrivons le réseau mis en place au CHI de Fréjus - St Raphaël pour la mise à disposition et la visualisation d'images médicales numériques issues de divers appareils de radiologie. L'aspect novateur réside dans l'adaptation de logiciels du domaine public à un parc d'ordinateurs PC basiques. Ce réseau unique en France (en 1998) permet, par une baisse sensible des coûts, d'envisager sa banalisation.

Ces travaux ont été développés avec **S. Nicolay** et **J. Michelis** dans le cadre de leur stage de DEA. Cette partie a donné lieu aux publications suivantes : [Ricordel 99, Nicolay 99, Puech 99, Puech 00, Michelis 00, Bouchouicha 00].

1.1 Conception d'un réseau pour la communication d'images médicales

Le standard DICOM (Digital Imaging and Communication in Medicine) [Nema 93] répond au besoin de communication des images numériques médicales. Ce dernier permet l'enregistrement d'images sur 4096 niveaux de gris (12 bits), échelonnés de -1000 à 3000 en unités Hounsfield. Elles contiennent également une entête avec des informations sur le malade, le type d'examen et la source utilisée (IRM, radio, scanner, ...). Mais les équipements nécessaires (matériels et logiciels) ont un coût prohibitif pour la plupart des centres hospitaliers et des cliniques. C'est dans ce contexte de rigueur financière que l'installation d'un réseau conforme à la norme DICOM, a été décidée au CHI de Fréjus - St Raphaël. Nous décrivons ici la conception de ce réseau pionnier n'intégrant que des logiciels du domaine public et des PC ordinaires.

1.1.1 Le standard DICOM et le service de radiologie du CHI de Fréjus-Saint-Raphaël

Le standard DICOM, pour la communication dans un environnement clinique, des images numériques médicales issues d'appareils de différents manufacturiers, s'imposait afin d'éviter les incompatibilités liées à la multiplication des formats propriétaires. Dès 1983 l'ACR (American College of Radiology) avec le NEMA (National Electrical Manufacturers Association), vite rejoints par d'autres organisations internationales de normalisation, ont formé un comité commun visant à développer une norme. En 1985 la version 1.0 du standard DICOM, pour un environnement réseau point à point, a été publiée. L'extension de systèmes de communication et d'archivage des images (ou PACS : Picture Archiving and Systems Communication) est visée, ainsi que la création de bases de données relatives aux diagnostics et interrogeables à distance. La version 3.0 actuelle du standard DICOM est parue en 1993 [Nema 93]. Elle propose notamment un protocole d'échange complet offrant une interface unique de communication pour des environnements réseaux standards tels que OSI (Open Systems Interconnect) ou TCP/IP (Transmission Control Protocol / Internet Protocol). Les règles de conformité au protocole DICOM ont été explicitement définies afin de structurer les niveaux de requêtes entre appareils, et d'identifier sans équivoque tout objet de l'information de l'imagerie médicale. Cette version 3.0 est aussi conçue de façon à intégrer les évolutions ultérieures de la norme. La structure hiérarchique du format DICOM se retrouve en pratique, au sein d'une base de données

identifiant séparément et dans l'ordre : le patient, les propriétés de l'examen, les séries de l'examen, les paramètres des images et enfin le chemin exact vers ces images. A chaque élément d'un tableau est associé un numéro d'identification unique, ainsi qu'un pointeur sur l'élément correspondant du tableau suivant. Le parcours de cette structure permet donc de reconstituer séquentiellement toute l'information relative à un examen.

Même si un certain nombre d'utilitaires de visualisation des images ou dédiés aux entêtes DICOM sont disponibles gratuitement, l'acquisition d'un réseau utilisant ce standard demeure inaccessible pour la plupart des centres hospitaliers du fait de son coût. Le Centre Hospitalier Intercommunal de Fréjus - St Raphaël compte 350 lits actifs. Depuis 1995, le service de radiologie est entièrement équipé de sources d'images numériques, leur conformité au standard DICOM a été assurée début 1998 :

- un tomodensitomètre (TDM ou scanner) produisant quotidiennement pour le CHI environ 480 images numériques de taille 512x512 et de profondeur 12 bits, soit 250 Mo/jour (sachant que le TDM n'est utilisé que pour moitié pour le CHI) ;
- une table de fluographie générant des images de taille 1024x1024 et de profondeur 8 bits. Environ 70 images sont générées par jour, représentant 70 Mo/jour ;
- un système à écran radioluminescent (ERLM), la taille des images est 1760x2370 et leur profondeur égale à 10 bits. Sont produites en moyenne 215 images/jour soit 1.8 Go/jour.

A l'origine les images numériques générées par un appareil de radiologie, demeuraient statiques sur le disque dur du calculateur propre à cette source. La mise à disposition des images s'effectuait sur supports argentiques. En 1998 le CHI s'est engagé dans un processus de communication des images numériques afin de pouvoir les visualiser à partir de n'importe quelle console, du service de radiologie dans un premier temps, de tout le CHI ensuite. Au-delà d'une consultation simple des images issues des diverses sources, ce système offre des modalités efficaces de recherche dans la base de données (par exemple à partir du numéro d'identification permanent du patient, ou d'une date d'examen). Ce réseau offre aussi toutes les garanties de confidentialité nécessaires, avec l'identification de l'utilisateur des consoles par mot de passe crypté.

1.1.2 Aspect général du réseau et ses composantes

La figure 2 donne une description synoptique du réseau mis en place. L'idée de base est de transférer, afin de stocker les images issues de chaque appareil de radiologie vers un serveur DICOM propre [Moore 94]. Le client peut cependant accéder directement à

l'ensemble des ressources image des bases de données. Cette architecture Client / Serveur complexe, car les serveurs sont distribués, a l'avantage de ne pas exiger la mise en oeuvre d'un super calculateur mais celle d'un ensemble de PC basiques reliés à un réseau local TCP/IP.

Dans l'optique de réduire encore les coûts, un autre objectif a été de déployer et d'adapter uniquement des logiciels appartenant au domaine public (licence GPL, General Public Licence) [Henri 97b] ou gratuits pour les établissements ayant des activités à but non lucratif.

L'architecture matérielle typique du PC mis en oeuvre est constituée d'un processeur Pentium MMX 233 Mhz, d'un ou deux disques de 6.5 Go, de 64 Mo de SDRAM et d'une carte réseau (fast Ethernet, 100 Mbits/s). La figure 2 fait apparaître trois types de serveurs. Tous fonctionnent avec le système d'exploitation LINUX (version RedHat 5.0, noyau 2.0.33). Ils se distinguent alors par leur application. Le serveur WWW, fonctionnant avec le logiciel APACHE 1.2, assure l'interrogation de la base de données et la mise à disposition en intranet des images médicales. Des scripts PHP/FI, générateurs de pages HTML dynamiques, permettent la consultation des images préalablement converties au format JPEG. Si le débit du réseau l'autorise il est également possible de rapatrier les images directement au format DICOM (c'est le cas dans le service de radiologie). Le serveur

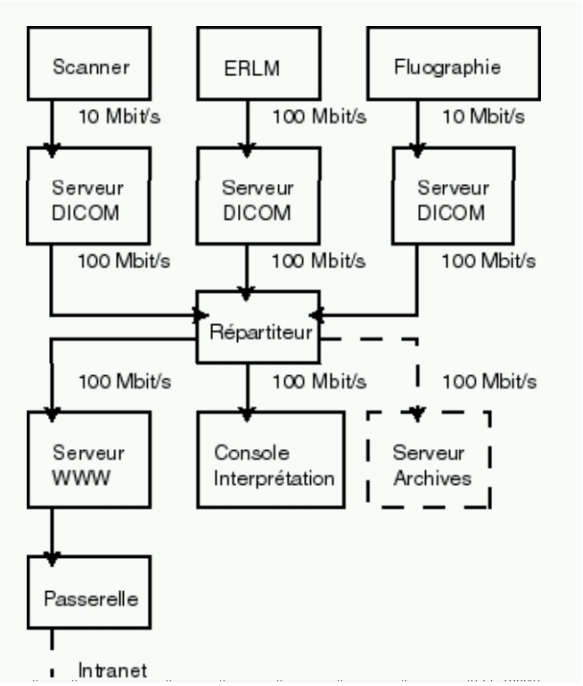


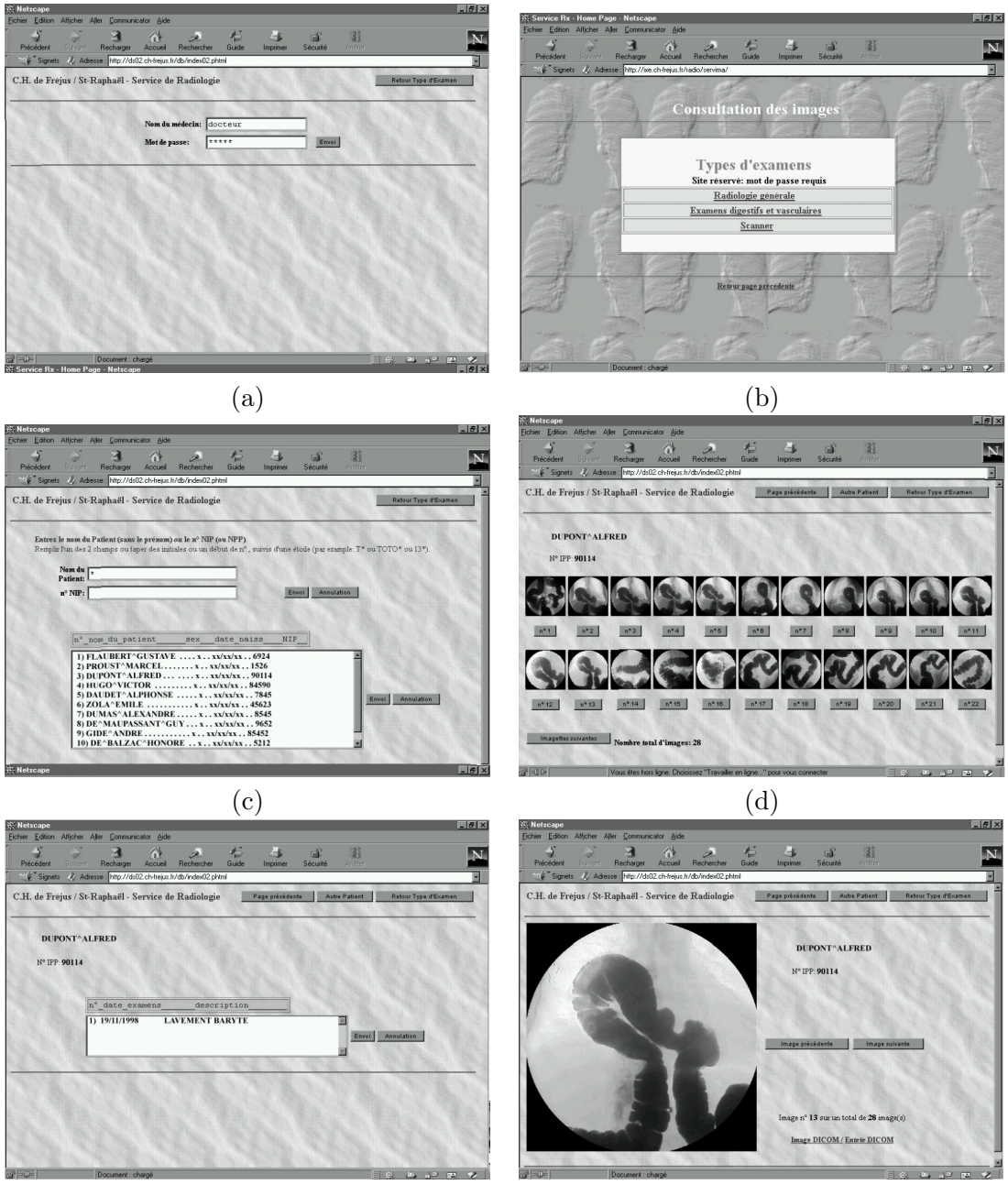
FIG. 2 – *Description synoptique du réseau*

DICOM, porte le logiciel serveur d'images DICOM : CTN (Central Test Node du Malinckrodt Institute of Radiology) [Henri 97a, Cox 98, Cox 97b] qui assure le stockage des images et s'appuie sur une base de données SQL (Structured Query Language) : mSQL. Un script PHP/FI gère automatiquement l'utilisation du disque en effaçant les données les plus anciennes s'il est plein. Le CTN de base fonctionnait avec le système d'exploitation UNIX, il a du être adapté par nos soins pour un usage sous LINUX avec mise en oeuvre de la bibliothèque graphique LESSTIF au lieu de MOTIF. Sur chaque serveur DICOM le logiciel APACHE est installé de façon à rapatrier, à la requête d'un client, les images sous forme comprimées. La compression du format DICOM à celui JPEG en mode sans perte est donc effectuée sur la machine cible (à titre d'exemple, pour une image scanner un taux de compression de 1:15 est typiquement obtenu). Le système de sauvegarde RAID (Redundant Array of Inexpensive Disks) n'a pas été installé en raison de son coût (dû aux cartes et disques supplémentaires), cet aspect n'est pas prioritaire car les données sont conservées à la source sur des bandes. Le serveur d'archives, assurant un stockage des données sur des bandes DLT (Digital Linear Tape) de 35 Go chacune, sera prochainement en fonction. Ces opérations de stockage seront pilotées automatiquement à l'aide de scripts PHP/FI. Avec l'ajout de ce dernier serveur, le réseau du CHI sera complet pour un fonctionnement sans film argentique. La figure 3 donne un exemple de navigation au sein du réseau [Ricordel 99]. Chacune des étapes, de l'identification de l'utilisateur à la visualisation d'une image particulière d'un examen, sont illustrées.

1.1.3 Conclusion et perspectives

Le projet de réaliser un réseau économique conforme au standard DICOM a été mené à bien. La difficulté consistait dans l'organisation de cette architecture distribuée et l'appareillement de logiciels GPL. L'installation du serveur d'archivages des images numériques médicales parachèvera le système. Ce réseau est en fonction au CHI de Fréjus - St Raphaël. Les responsables des autres centres hospitaliers et cliniques peuvent ainsi venir le visiter, et s'en inspirer pour la mise à niveau de leurs propres installations.

La mise en place du réseau pour la communication et la visualisation des images numériques a été effectuée au niveau du service de radiologie, et son extension en intranet à tout le CHI est achevé. L'évaluation pour une montée en charge du système doit à présent être validée, ainsi que la prise en main par les médecins de l'outil de consultation (le navigateur Netscape 4.5). Cependant cette première structure opérationnelle fait



(a) (b)

(c) (d)

(e) (f)

FIG. 3 – *Exemple d'une procédure de navigation au sein du réseau. a) Identification de l'utilisateur, b) Choix du serveur, c) Choix du patient et de l'examen, d) Visualisation de l'examen, e) Visualisation d'une image de l'examen, f) Lecture de l'entête de l'image.*

déjà du CHI un centre pilote unique en France. L'ajout du serveur d'archives permettrait même un fonctionnement sans film, et rendrait nul le risque de pertes de clichés. De plus les économies ainsi faites garantiraient la rentabilité des investissements engagés, particulièrement modestes en l'occurrence[1].

1.2 Visualisation d'images haute résolution au travers d'un arpenteur : application à l'imagerie médicale

Nous présentons dans ce travail une application Client/Serveur pour la visualisation d'images médicales issues de divers appareils de radiologie. Ces images sont stockées sur un réseau conforme à la norme DICOM au Centre Hospitalier Intercommunal (CHI) de Fréjus-Saint-Raphaël. L'originalité de ce travail réside dans le fait que l'opérateur n'a besoin que d'un navigateur Web pour pouvoir visualiser des images hautes résolution lui permettant d'avoir une aide précieuse pour son diagnostic et ceci de n'importe quel ordinateur connecté au réseau.

L'objectif de ces travaux, est de permettre la consultation ainsi que la visualisation d'images stockées sur un serveur DICOM. En effet, la visualisation de ce genre d'image nécessite des visualiseurs performants, vu la taille de celles-ci qui est comprise entre 512×512 et 1024×1024 pixels, avec un codage sur 12 bits de niveau de gris par pixel. Pour palier à ce problème, nous proposons une application Client/Serveur qui permet de consulter une base de données images à partir d'un simple navigateur Web. Des travaux permettant la consultation de bases de données en ligne sont d'actualité [Sclaroff 99, Florescu 00].

Nous avons présenté dans la section 1.1 le contexte de notre travail avec une description du réseau du CHI de Fréjus-Saint-Raphaël. Dans cette section nous allons décrire les différents langages utilisés lors du développement de notre application, ainsi que l'interface de visualisation des images médicales.

1.2.1 Interactivité du réseau DICOM du CHI

Le client peut accéder directement à l'ensemble des ressources image des bases de données. La figure 3 illustre un exemple de navigation au sein de ce réseau.

La confidentialité est obtenue grâce à la structure hiérarchique du format DICOM. Celle-ci se retrouve en pratique, au sein d'une base de données identifiant séparément et dans l'ordre : le patient, les propriétés de l'examen, les séries de l'examen, les paramètres

1. Nos remerciements vont au Conseil Général du Var pour l'intérêt marqué vis à vis de ce travail et l'aide apportée

des images et enfin le chemin exact vers ces images. A chaque élément d'un tableau est associé un numéro d'identification unique, ainsi qu'un pointeur sur l'élément correspondant du tableau suivant. En parcourant cette structure, on peut reconstituer séquentiellement toute l'information relative à un examen.

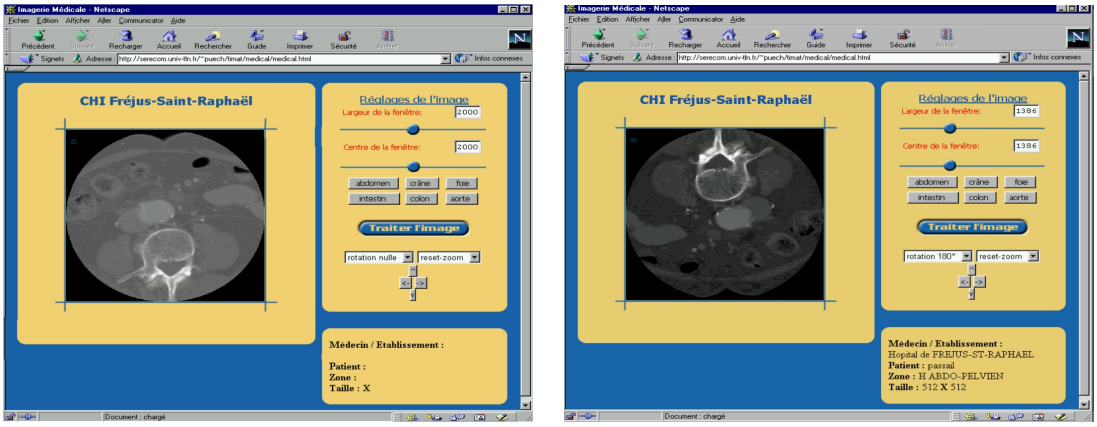
Le développement de notre application a nécessité l'utilisation de codes HTML et PHP pour la partie formulaire et interrogation du serveur, et Javascript pour les actions et événements du coté client. Le Javascript est aussi utilisé pour l'aspect graphique. Le langage de programmation PHP est un langage de script interprété dans les pages HTML [Lacroix 00], déclenché par les machines clientes et traité par le serveur. Il permet de construire dynamiquement sur le serveur des pages HTML à destination d'un client contenant les résultats de calculs ou de requêtes SQL adressés à un système de gestion de bases de données (SGBD). Il peut s'interfacer avec la quasi totalité des SGBD du marché, qu'ils soient commerciaux ou qu'ils viennent du monde du logiciel libre. PHP3 est distribué librement et gratuitement sous la licence GNU GPL.Dans le cadre de notre application le code PHP permet de déclencher un programme exécutable en langage C qui convertit une image DICOM en JPEG avec les réglages voulus. L'interrogation pour l'accès aux données se fait à l'aide de requêtes SQL.

Pour les actions déclenchées par des évènements nous avons utilisé le langage Javascript. Le choix de ce langage a été motivé par sa facilité d'utilisation, ainsi que le fait qu'il soit interprété par les navigateurs Web à partir des machines clientes. Notons que le manque de standardisation des fonctions Javascript entre les différents navigateurs a aussi été pris en compte.

1.2.2 Description et utilisation de l'interface graphique de visualisation

Notre application repose essentiellement sur l'utilisation d'une interface, présentée figure 4.a, par l'intermédiaire d'un arpenteur [Bouchouicha 00]. Le développement de cette interface a été effectué avec des logiciels appartenant au domaine public (licence GPL), ou gratuits afin de réduire les coûts.

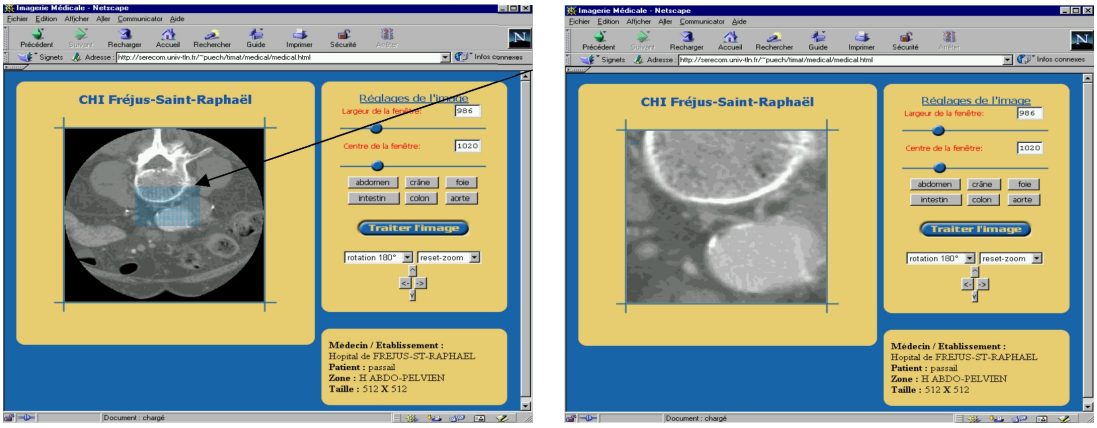
L'interface de visualisation dispose d'une zone d'affichage sur la gauche. Cette zone est prévue pour afficher les images DICOM converties en JPEG dans une fenêtre de 300×300 pixels. Sur la droite, l'utilisateur dispose d'un menu avec des boutons d'action et des menus déroulants. Ces derniers permettent de régler certains paramètres pour la conversion et de déclencher le traitement. Au chargement d'une image, des réglages standards prédéfinis sont appliqués.



(a) (b)

FIG. 4 – *a) Menu général, b) Exemple de rotation.*

La simplicité de l'utilisation de cette interface, ainsi que sa convivialité, permettent à l'opérateur de choisir une image, qui sera affichée dans un premier temps en basse résolution dans une fenêtre de 300×300 pixels. Ensuite, l'opérateur peut fixer différents paramètres, lui permettant de faire apparaître certaines zones d'intérêt de l'image, tels que la largeur de la fenêtre de niveaux de gris ainsi que son centre. Ces deux paramètres sont nécessaires pour la conversion du format DICOM codé sur 4096 niveaux de gris vers le format d'affichage JPEG qui lui n'est codé que sur 8 bits en profondeur de niveaux de gris.



(a) (b)

FIG. 5 – *a) Sélection d'une zone, b) Agrandissement d'une zone sélectionnée.*

D'autre part, l'interface présente des boutons dont le réglage est prédéfini, permettant ainsi à l'opérateur d'avoir un affichage correspondant à des parties anatomiques telles

que l'aorte, l'intestin ou le foie. Deux fonctions supplémentaires viennent enrichir cette interface, la rotation, représentée figure 4.b et l'agrandissement par zoom, figures 5.a et b. Dans le cas d'un zoom il est alors possible d'obtenir la haute résolution de l'image originale au format DICOM.

Soulignons que tous les traitements sont appliqués sur l'image haute résolution du coté serveur. L'image ainsi traitée est ensuite convertie au format JPEG afin d'être transférée puis visualisée au travers d'un arpenteur.

1.2.3 Conclusion et perspectives

Après une phase d'évaluation, nous avons montré qu'il est possible d'accéder à des images haute résolution quasiment en temps réel. Ces zones d'images haute résolution sont créées dynamiquement en interrogeant le serveur. La taille des pages HTML contenant ces images reste donc très petite. L'interface développée est simple d'utilisation et permet à l'opérateur d'avoir une aide au diagnostic plus fine.

La mise en place de l'interface de visualisation ainsi que la prise en main par le médecin ont été effectuées avec succès. Cependant, l'ajout de nouveaux outils de traitement d'images via le Web permettrait d'enrichir cette interface, qui formerait un outil incontournable pour aider l'opérateur dans le diagnostic de certaines pathologies [Tabaty 00]. D'autre part, il est envisageable d'obtenir une visualisation tridimensionnelle d'organe, et ceci en utilisant des séquences d'images tomodensitométriques [Michelis 00]. Ceci permettrait d'avoir une information supplémentaire, facilitant la tâche de l'opérateur dans le cadre du diagnostic. Enfin, il serait souhaitable pour des raisons de confidentialité de sécuriser l'accès à la base de données images. Une solution, serait l'utilisation des techniques de tatouage d'images ou watermarking [Ruanaidh 96]. En effet, il nous paraît intéressant d'insérer dans l'image JPEG les mêmes informations textuelles contenues dans le format DICOM. De cette manière, le praticien pourra alors à tout moment réinjecter, dans l'application en ligne, une image JPEG, traitée et sauvegardée sur son poste client, afin de récupérer les informations concernant le patient. Nous envisageons aussi de tatouer l'image afin de garder en mémoire les caractéristiques des traitements effectués.

1.3 Intégration d'applets JAVA dans un réseau d'images médicales : aide au télé-diagnostic

Dans le cadre du projet STIMAT (Système de Traitement des Images Médicales Aide au Télé-diagnostic) nous cherchons à répondre aux besoins des médecins spécialistes dans le domaine de la radiologie et plus précisément du radiodiagnostic. Ce projet offre la possibilité d'utiliser les techniques de communication informatique actuelles associées à des méthodes de traitement d'images. Nous présentons une vision de la télé-médecine au travers du développement d'un outil de télé-diagnostic. Elle représente un caractère indispensable à la médecine moderne dans le suivi des patients. De plus, actuellement le diagnostic utilise des représentations 3D des organes qui donnent des informations supplémentaires par rapport aux images en coupe. Ces reconstructions 3D sont généralement obtenues à partir de traitements classiques, comme le seuillage et la soustraction d'images, souvent longs et fastidieux. A partir de séquences d'images médicales, il nous apparaît possible de mettre en oeuvre des techniques plus performantes basées sur des méthodes de contours actifs pour la détection des organes. L'analyse des méthodes existantes et la proposition de nouvelles méthodes seront détaillées dans le chapitre 2. Nous proposons ici d'implémenter ces méthodes classiques de manière à pouvoir les exécuter à distance depuis un navigateur Internet. Dans le cadre de nos travaux nous avons développé un paquetage en JAVA dédié à la détection et à la reconstruction 3D de l'aorte. Une station de travail spécifique du CHI de Fréjus-St-Raphaël, isolée du réseau interne, permet aux médecins spécialistes de traiter les images au format DICOM obtenues par numérisation depuis divers appareils de radiologie. Nous proposons de développer une méthode de détection de contours d'organes directement réalisée sur les séries d'images stockées sur le serveur DICOM. Outre l'aspect traitement d'images, le développement informatique représente une partie importante de la mise en place d'un outil de télé-diagnostic. Ainsi, le langage JAVA est un des candidats les plus prometteur par ses propriétés comme la portabilité, le caractère client/serveur et la relative facilité à mettre en place une interface graphique au travers d'un arpenteur grâce aux applets. Nous présentons la base des données à exploiter qui est composée d'images médicales au format DICOM. Nous présentons également la topologie du réseau du CHI de Fréjus-St-Raphaël où l'applet sera implantée. Nous développons ensuite le traitement des images médicales en utilisant les contours actifs [Latombe 97, Abrantes 93]. Outre la description des différents éléments qui ont permis l'élaboration des algorithmes de détection d'objet, nous apportons une solution pour propager la détection au travers d'une séquence

d'images médicales. Les résultats obtenus par l'applet montrent que le traitement d'images est possible au travers d'un réseau, et de plus, avec des machines de type PC de faible capacité consacré habituellement aux tâches de bureautique. Finalement, nous présentons le traitement des images médicales et l'implémentation des algorithmes en JAVA.

1.3.1 Réseau et DICOM

Le format DICOM fournit d'une part l'image numérique et d'autre part une information texte relative à l'examen effectué. L'image est alors codée sur plus de 4000 niveaux de gris par pixel. Il est possible de mettre en évidence les zones que le médecin spécialiste désire analyser en ne gardant qu'une partie de l'information haute résolution. Dans le cas de tests sur la détection de l'aorte, nous avons centré notre fenêtre de largeur 400 niveaux de gris sur 20 unités Hounsfield. La figure 6 montre la mise en évidence de l'aorte.

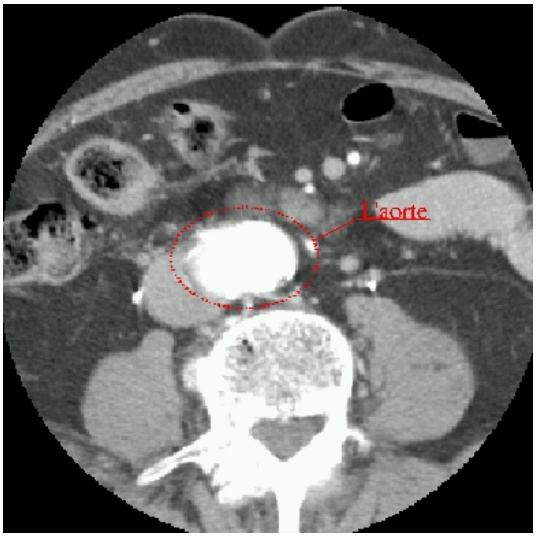


FIG. 6 – *Mise en évidence de l'aorte.*

L'exploitation d'une telle quantité de données représente un travail lourd et complexe à mettre en oeuvre sur un réseau où les utilisateurs de l'outil de télé-diagnostic se connectent avec un navigateur Internet. Par conséquent, dans le cadre du projet STIMAT, un script PHP associé à un serveur a été produit pour transformer le format DICOM en JPEG non destructif [Bouchouicha 00]. Le JPEG ayant l'avantage d'être approprié au réseau est de plus exploitable dans un programme JAVA.

Le descriptif, figure 2, montre l'environnement matériel dans lequel l'applet JAVA est intégrée. L'applet JAVA se situe sur le serveur WEB. A travers la passerelle d'accès,

les médecins peuvent, tout d'abord s'identifier, puis accéder aux clichés de leurs patients [Ricordel 99]. Avec l'applet JAVA, les médecins spécialistes en traitement d'images, peuvent créer une base de données contenant les reconstructions 3D consultables en intranet de la même manière que les clichés.

1.3.2 Traitement des images médicales

En premier lieu, nous présentons les méthodes envisagées pour détecter et reconstruire les organes en 3D. Pour cela, il nous faut décrire les éléments implémentés dans l'applet java. A l'issue de l'examen, nous disposons d'une séquence de clichés au format DICOM. La détection de contour d'objet représente la première étape du développement de notre méthode. Elle est basée sur la technique des contours actifs classiques ou "snake" [Cocquerez 95]. Les caractéristiques de cette méthode nous orientent vers la création d'objets. La première caractéristique est l'initialisation du contour actif à l'intérieur de la région à détecter. Le contour actif fermé forme ainsi un polygone. Par conséquent, le nombre de segments ou de sommets de celui-ci traduit la qualité de résolution du contour final. Ainsi la géométrie de la région peut demander un effort sur la précision surtout dans des zones de forte courbure. Pour cela, l'énergie interne traitant des contours actifs [Cohen 91] est le caractère principal dans le ré-échantillonnage de notre polygone. L'expression utilisée dans notre cas est la suivante :

$$K = |sin(\phi)|, \tag{1}$$

où K est la courbure associée à un point de notre contour actif, et ϕ est l'angle entre deux segments. L'énergie externe est le critère de détection du contour de la région en question. En effet, les sommets du polygone composant notre contour actif, se confondent avec l'objet à détecter lorsque l'énergie externe est maximale. Son expression est la suivante :

$$P(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-1}{2}\frac{x^2+y^2}{\sigma^2}} \|\nabla I(\vec{x},y)\|, \tag{2}$$

où l'énergie externe P regroupe dans son expression une gaussienne d'écart type σ et la norme euclidienne du gradient de l'image.

La détection doit alors être propagée dans toute la séquence d'images. Pour cela, il s'agit d'automatiser le passage d'une image à l'autre. En utilisant le contour trouvé sur la coupe k, il est possible d'initialiser la détection sur l'image k+1. En effet, tout en respectant les conditions évoquées précédemment sur l'initialisation des contours actifs, la détection d'objet peut être relancée en prenant compte des informations recueillies à l'étape précédente. Par conséquent, l'intervention humaine est réduite puisque seule la

première image doit être initialisée. Le résultat de ces deux étapes nous permet de replacer l'ensemble de ces points dans un espace tridimensionnel afin d'obtenir une visualisation 3D de l'organe.

Soulignons tout d'abord l'impact du ré-échantillonnage. Le but de celui-ci est d'améliorer la qualité topographique [Delingette 00] du contour actif pendant la phase d'évolution dans l'image et au moment où il se confond avec le bord de l'objet. Les sommets du polygone sélectionnés manuellement à l'intérieur de l'aorte sont représentés figure 7.a. Après validation, un ré-échantillonnage est effectué avant même une progression du contour actif, figure 7.b. Le ré-échantillonnage est fait à chaque fois que le contour actif évolue jusqu'à la détection complète. Notons que le critère de courbure influe sur le nombre de points ajoutés.

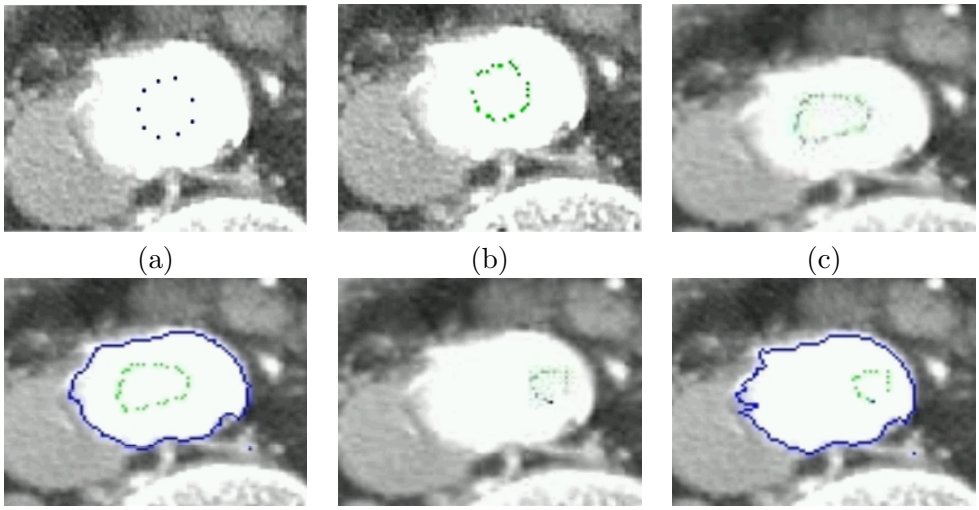


FIG. 7 – *a) Sélection des germes, b) Ré-échantillonnage, c) Initialisation, d) Détection du contour de l'aorte, e) Initialisation différente, f) Nouvelle détection du contour de l'aorte.*

Après ré-échantillonnage et progression de notre contour actif, l'objet à détecter est isolé avec une certaine précision. Les figures 7.d et 7.f montrent aussi que la précision dépend, pour certaines zones de l'image, du contour initial des figures 7.c et 7.e correspondantes. Il est donc important de s'intéresser à la forme initiale du contour actif et d'optimiser la détection par la recherche du maximum de l'énergie externe. Le ré-échantillonnage montre son efficacité sur le fait que le contour actif épouse de manière très élastique les formes de l'aorte.

1.3.3 Outils 3D de télé-diagnostic

Selon les données à exploiter, il est possible d'utiliser des méthodes de reconstruction 3D similaires à celles utilisées en 2D. La segmentation dans des objets 3D peut s'obtenir par minimisation de l'énergie d'une surface déformable [Cohen 92]. La reconstruction 3D peut être aussi effectuée par la technique du MIP (Maximum Intensity Projection). Ainsi, la création d'objet 3D s'exécute soit à partir de données 3D, soit par reconstruction à partir d'images 2D. En gardant seulement les coordonnées des points qui forment le contour de l'aorte, nous disposons d'un squelette 3D de l'aorte.

Nous avons classifié les objets en deux catégories [Michelis 00]. La première correspond à la détection du contour de l'aorte. La seconde permet la reconstruction 3D de manière explicite. Avant de détailler ces deux catégories nous vous présentons l'architecture des classes et les objets créés représentés figure 8. Nous avons ainsi les objets suivants :

- les "Germes" qui se déploient à l'intérieur du contour à détecter. Ils possèdent une grande quantité d'informations afin que leurs progressions s'arrêtent sur le contour de l'aorte,
- les "Contours" détectés. Nous associons à chacun d'entre eux l'image traitée correspondante. Un dialogue permanent entre les objets "Contour" et "Germes" appartenant à celui-ci permet une mise à jour des informations liées aux "Germes" et leur environnement,
- la "Séquence" représente l'ensemble des contours trouvés. Explicitement, elle nous donne la structure 3D de l'aorte,
- enfin les objets "Arête", "Coordonnées" et "Surface Active" sont des intermédiaires pour les calculs et le positionnement des "Germes".

Une telle structure nous a permis de bien partager les fonctionnalités à chacun des objets. Nous avons donc bien deux catégories d'objets que nous allons détailler. La première catégorie concerne la détection du contour. Pour cela les objets "Coordonnées", "Germes" et "Arêtes" ont un rôle prépondérant. En effet, ces objets établissent la structure du contour à détecter. L'objet "Germe" dispose de toutes les informations telles que l'énergie ou le gradient de l'image dans la zone où il se trouve. Toutefois, il fait appel à l'objet "Contour" pour disposer d'une certaine indication comme la direction de propagation et les niveaux de gris relatif à sa position dans l'image. L'objet "Arête" lie les "Germes" entre eux, tandis que l'objet "Coordonnée" caractérise le positionnement du "Germe". La

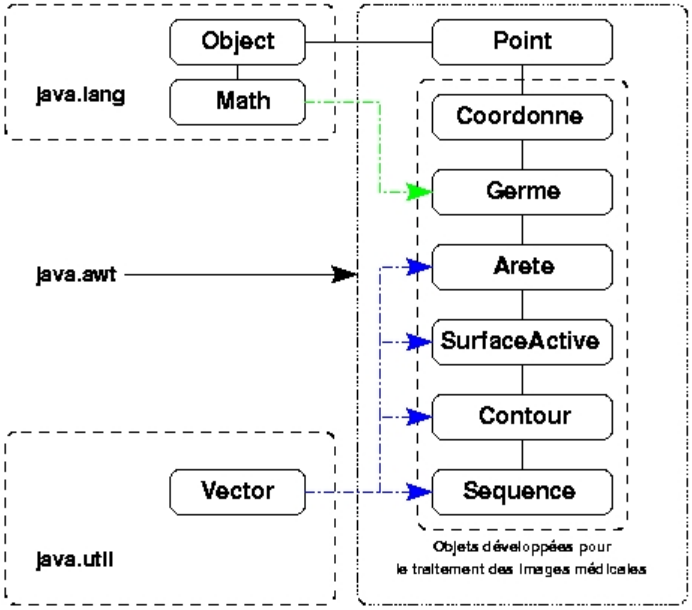


FIG. 8 – *Mise en place d'un paquetage JAVA permettant la propagation de contour dans une séquence d'images. Descriptif synoptique du réseau.*

seconde catégorie d'objets recueille toutes les données fournies par les objets "Germes" qui sont stockées dans chaque objet "Contour". C'est le rôle de l'objet "Séquence" qui constitue un tableau de positionnement de points afin de créer un modèle de l'aorte en trois dimensions. Le but essentiel de la "Surface Active" est d'isoler les points appartenant à l'intérieur du contour. La seconde catégorie d'objets est constituée donc de trois objets "Surface Active", "Contour" et "Séquence". Toutefois l'objet "Contour" est un objet de transition dans le sens où les "Germes" ont besoin d'interroger en permanence celui-ci pour mettre à jour certaines données nécessaires aux calculs d'énergie et de positionnement initial. Ce paquetage étant destiné au développement, soit d'une applet, soit d'une application de type client/serveur, est composé de six classes chacune représentative d'un objet, figure 8.

1.3.4 Conclusion et perspectives

Le traitement des images médicales fait appel à des notions ayant un rôle à jouer dans les différents stades du développement de l'outil de télé-diagnostic. Ainsi, la topologie et la géométrie permettent par le ré-échantillonnage du contour actif d'obtenir une meilleure résolution de celui-ci. D'autre part, le calcul du gradient fixe l'énergie externe de chacun

des germes. Cette énergie nous indique à chaque déplacement des germes, si la position de ceux-ci appartient au contour final. Grâce à la détection par contour actif, la segmentation de l'image est effectuée en isolant l'objet désiré. La finalité de ces diverses manipulations est la reconstruction 3D de l'objet. L'élaboration d'un outil de télé-diagnostic demande de la rigueur aussi bien dans le traitement des images que dans les méthodes de programmation algorithmique. Le secteur médical est très intéressé par de tels outils pour le télé-diagnostic. En effet, la télé-médecine est en pleine croissance car l'aspect réseau apporte aux médecins plus de facilités et de rapidité dans l'établissement d'un diagnostic. Le format DICOM a été le premier pas vers la médecine moderne. La télé-médecine est l'avenir d'une médecine plus efficace et plus rapide. La création de services informatiques avec des aspects réseau de type client/serveur abonde dans ce sens. Nous pourrions envisager que les médecins spécialistes supervisent et contrôlent le travail effectué par un serveur afin de garantir une bonne interprétation du diagnostic du côté client. Enfin, l'étude et l'élaboration d'un tel service peuvent amener à un produit complet utilisable soit dans les centres radiologiques, soit dans les universités de médecine comme outil pédagogique. Il nous est donc intéressant de prolonger le développement d'outils 3D pour le télé-diagnostic. Ainsi nous pourrions continuer sur une optimisation de la propagation du contour dans une séquence d'images médicales afin d'obtenir un rendu 3D. Il s'avère nécessaire de créer une visionneuse 3D adaptée aux besoins des médecins spécialistes et à l'environnement réseau. En effet, l'application doit être développée sur le modèle client/serveur, afin de sécuriser et de gérer correctement la base de données DICOM. Nous avons également développé ce type d'application au niveau du CHITS (Centre Hospitalier Interurbain de Toulon-la-Seine sur Mer.

Chapitre 2

Détection de contours et reconstruction 3D

Dans ce chapitre, je présente mes travaux sur la détection et le suivi de contours d'objets déformables, pour l'étude d'une séquence de coupes issues d'un appareil tomodensitomètre à rayons X. Dans un premier temps, nous avons analysé les méthodes utilisées dans les services d'imagerie médicale. Elles sont principalement basées sur des techniques de seuillage et de soustraction d'images. Nous avons d'abord proposé d'améliorer cette méthode en réalisant semi-automatiquement l'extraction d'une seule structure anatomique ciblée. Notre méthode vise à mettre en oeuvre un modèle de contours actifs et procède en trois étapes : la détection d'un contour sur la première coupe, la propagation de ce contour en le déformant aux coupes connexes et la reconstruction 3D.

Dans ce chapitre, nous présentons également un nouvel algorithme de suivi d'un organe dans une séquence d'images médicales afin de réaliser une reconstruction 3D. La méthode automatique que nous proposons permet de suivre le contour externe d'un organe anatomique dans toute la séquence à partir d'un contour initialisé par l'utilisateur sur la première image. Les opérations nécessaires pour notre méthode de suivi s'appuient sur une segmentation par contours actifs basée région. La localisation des objets avec une prédiction dynamique de déplacement est basée sur les fonctions de courbes de niveaux et sur la définition de région d'intérêt pour l'estimation locale robuste du modèle de l'image. Une application de cette méthode est la reconstruction 3D de l'aorte abdominale.

Dans la section 2.1 nous présentons une analyse et des améliorations de méthodes de reconstruction 3D de l'aorte à partir d'une séquence d'images tomodensitométriques. Nous détaillerons, section 2.2, une méthode permettant de propager automatiquement des contours actifs dans une séquence d'images médicales.

Ces travaux ont été développés avec **G. Ledanff** dans le cadre de son stage de DEA et avec **K. Djemal** dans le cadre de sa thèse. Cette partie a donné lieu aux publications suivantes : [Nicolay 99, Puech 99, Puech 00, Djemal 02, Djemal 03b, Djemal 03a, Djemal 04, Djemal 05].

2.1 Analyse et amélioration de méthodes de reconstruction 3D de l'aorte à partir d'une séquence d'images tomodensitométriques

Dans un premier temps nous analysons les méthodes existantes dans les services d'imagerie médicales. Elles reposent principalement sur des techniques de seuillage de niveau de gris. Les contours issus de l'ensemble des coupes de la série sont ensuite utilisés afin de reconstruire en 3D par interpolation l'objet anatomique. Des comparaisons de méthodes manuelle et automatique ont déjà été réalisées pour effectuer une segmentation 3D, [Jayaraman 97, Udupa 97].

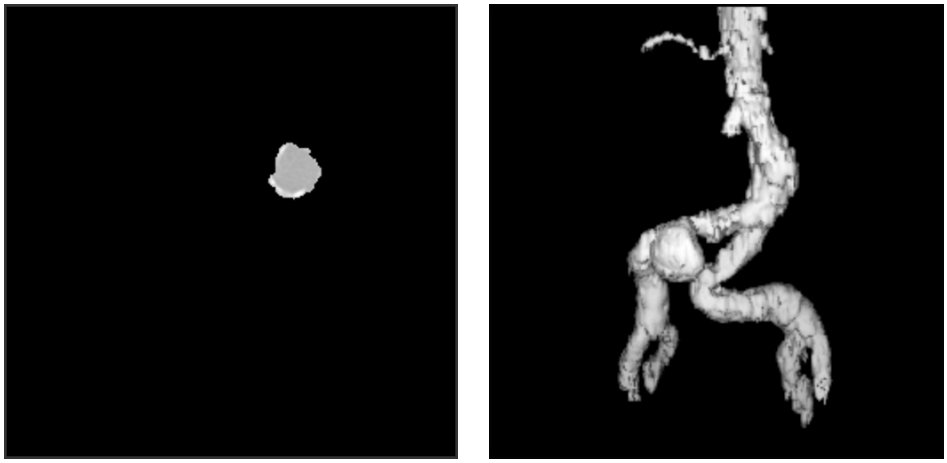
La méthode d'amélioration proposée met en oeuvre des techniques de contours actifs, [Ducottet 97, Kass 88]. Celle-ci procède en trois étapes. La première étape est la détection d'un contour sur une première coupe. Ce processus de segmentation nécessite une initialisation par sélection de points, [Cocquerez 95]. A partir de ce contour initial, dans la deuxième étape, nous montrons comment propager ce contour en le déformant aux coupes connexes. La dernière étape concerne alors la reconstruction 3D, [Gao 97, Fiebich 97].

Dans première partie nous détaillons la méthode la plus souvent utilisée dans les services de radiodiagnostic. Nous montrons ensuite un approfondissement de la méthode précédente pour un examen de qualité. Enfin nous présentons une méthode semi-automatique de détection de contours d'une aorte [Nicolay 99, Puech 99, Puech 00]. Pour terminer nous développons une amélioration des techniques de détection présentées précédemment.

2.1.1 Seuillage élémentaire et sélection de l'aorte

Les services d'imagerie médicale utilisent le plus souvent des consoles dédiées à la reconstruction bidimensionnelle des images acquises. La reconstruction 3D est effectuée à partir d'une séquence de coupes issues, pour ce qui nous concerne, d'un TDM. Ces coupes contiennent l'aorte, objet à identifier, mais également diverses structures anatomiques gênantes comme la colonne vertébrale.

Pour visualiser l'aorte et ses principales branches collatérales et terminales, il faut effectuer un seuillage de niveaux de gris de l'image originale. Le format normalisé en imagerie médicale, est le format Dicom qui est codé sur 4096 niveaux de gris. Pour notre examen, le seuillage le plus performant est effectué entre 160 et 1374. Cette manipulation engendre une perte d'information considérable. Mais grâce à ce seuillage les contrastes des images seront plus marqués et permettront ainsi d'effectuer un traitement. Une sélection effectuée simplement sur la zone concernée, permet de ne garder que les parties connexes à cette sélection. Cette méthode a permis d'isoler l'aorte, figure 9.b, du reste des structures anatomiques. La figure 9.a. est l'une des coupes 2D après seuillage de l'aorte qui permet la reconstruction de la figure 9.b. Cette méthode est simple mais elle est fortement destructrice. De plus la recherche du seuil le plus pertinent, permettant cette sélection, est longue et répétitive.



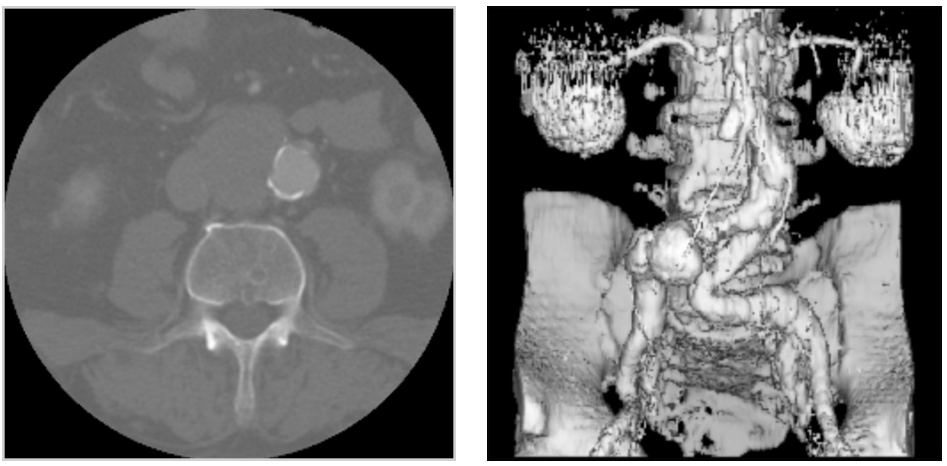
(a) (b)

FIG. 9 – *a) Une coupe 2D de l'aorte sélectionnée directement, b) Vue 3D de l'aorte après sélection directe par simple seuillage.*

Du fait de ces résultats non performants, actuellement des méthodes approfondies sont utilisées dans les services de radiodiagnostic. Elles sont basées sur une soustraction de la colonne vertébrale avant sélection de l'aorte.

Durant la reconstruction 3D la visualisation est souvent réalisée en MIP (Maximum Intensity Projection), mais pour des facilités de visualisation celle-ci sera présentée en rendu surfacique, figure 10.b. Un exemple d'une vue en MIP est représenté figure 12.b. La figure 10.a. est l'une des coupes 2D permettant la reconstruction 3D de la figure 10.b. La méthode se décompose en trois parties : sélection de la colonne vertébrale après seuillage,

dilatation de la colonne vertébrale, soustraction et visualisation finale.



(a) (b)

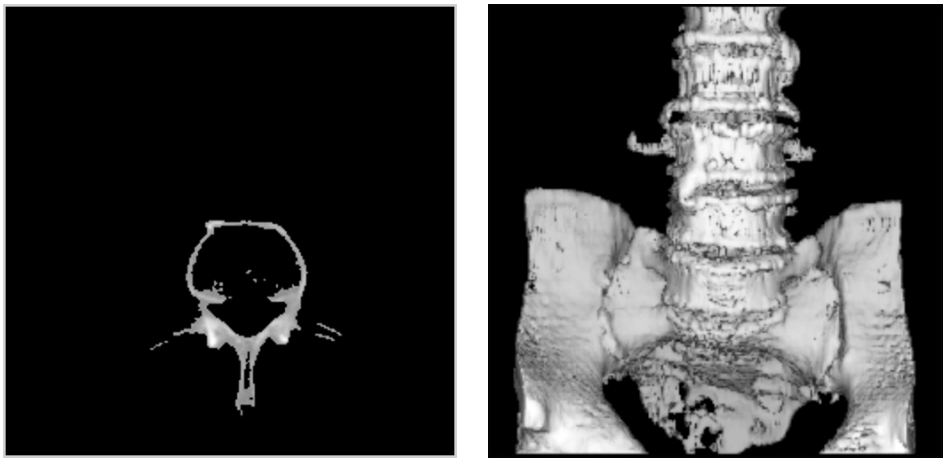
FIG. 10 – *a) Une coupe 2D originale contenant l'aorte et la colonne vertébrale, b) Visualisation 3D d'ensemble de l'aorte et de la colonne vertébrale obtenue par simple seuillage.*

2.1.1.1 Sélection de la colonne vertébrale après seuillage

Un seuillage reste indispensable dans cette méthode pour isoler la colonne vertébrale. Après sélection de celle-ci, seules les parties connexes sont conservées. Cet objet sera soustrait des données initiales de la manière la plus complète, en évitant au maximum de concerner les structures avoisinantes, appartenant à l'aorte. Les résultats de cette étape sont présentées figure 11.b. avec une coupe 2D, figure 11.a. Tant que des zones de contact entre la colonne vertébrale et l'aorte subsistent, il faut élever le seuil jusqu'à leurs suppressions.

2.1.1.2 Dilatation de la colonne vertébrale

Malheureusement, un seuillage préalable souvent important est responsable d'une dégradation de l'image de la colonne vertébrale et d'une érosion de son volume apparent, figure 11.b. Une méthode de correction consiste à appliquer des techniques de morphologie mathématique [Cocquerez 95]. Une dilatation puis une érosion avec un coefficient donné sont effectuées sur la colonne vertébrale, obtenue par la manipulation décrite section 2.1.1.1.

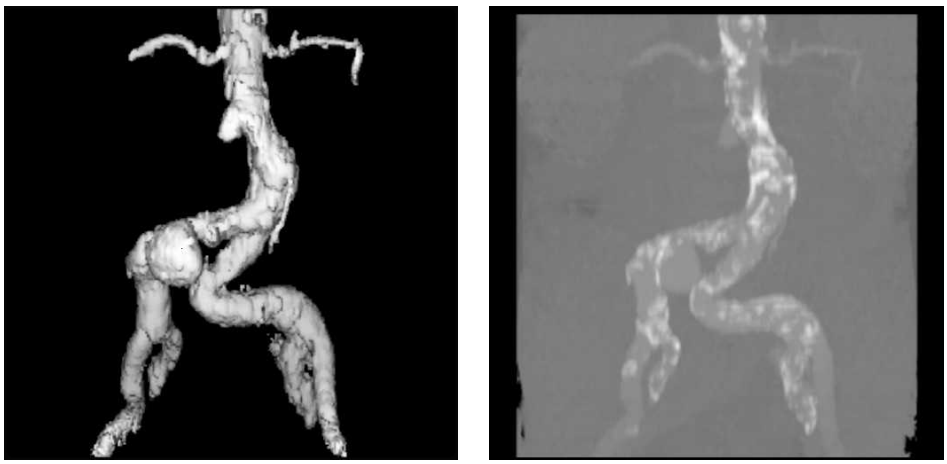


(a) (b)

FIG. 11 – *a) Coupe 2D de la colonne vertébrale sélectionnée, b) Vue 3D de la colonne vertébrale après sélection.*

2.1.1.3 Visualisation pour l'examen final

Le volume obtenu est alors soustrait de l'image native, figure 10.a. Par définition la vue en MIP montre les pixels d'intensités maximales. Les pixels correspondants à la colonne vertébrale viennent dans ce cas se mélanger à ceux de l'aorte, il convient donc de la soustraire au préalable. La figure 12.a est le rendu surfacique de l'aorte obtenue par l'ensemble de ces manipulations, la figure 12.b est sa vue en MIP. Le passage de la vue surfacique à la vue en MIP (Maximum Intensity Projection) s'effectue sans aucune difficulté par l'ensemble des consoles dédiées à l'imagerie médicale.



(a) (b)

FIG. 12 – *a) Vue en rendu surfacique de l'aorte, b) Vue en MIP de l'aorte.*

Il est possible d'évaluer qualitativement le gain d'informations apporté par l'ensemble des manipulations longues et répétitives en comparant les figures 9.b et 12.a. La dilatation préalable de la colonne vertébrale est malheureusement souvent responsable d'un recrutement de pixels n'appartenant pas à celle-là. La soustraction effectuée sera alors excessive et dégradante pour cet examen.

2.1.2 Détection des contours d'une aorte à partir d'une méthode semi-automatique

Des méthodes semi-automatiques apportent un gain de temps pour l'utilisateur, [Fiebich 97]. Des comparaisons de méthodes manuelle et automatique ont déjà été réalisées pour effectuer une segmentation 3D, [Jayaraman 97, Udupa 97]. Dans le cas de notre étude, la méthode semi-automatique utilisée n'apporte pas souvent de bon résultat. Cette méthode utilise une technique de seuillage pour effectuer sa recherche. Elle consiste à sélectionner une zone de la première image de la série récupérée par le TDM et de poursuivre cette zone sur l'ensemble de la série. La manipulation, effectuée par l'utilisateur, se décompose en deux parties : la sélection de la zone à poursuivre et le lancement de la poursuite de contours.

2.1.2.1 Sélection de la zone avec ou sans recherche de contour

La sélection de la zone à poursuivre doit être effectuée manuellement avec sans recherche du contour le plus proche ou avec recherche automatique du contour le plus proche en temps réel. Sans recherche, il sélectionne des points à l'aide de la souris qui seront immédiatement reliés par des segments. Mais la zone sélectionnée, n'est pas parfaite ; les risques d'avoir un résultat peu précis augmentent. Avec recherche automatique, il s'expose à un risque de débordement de la zone de sélection voulue. En effet cette technique recherche la zone de transition la plus proche. Si cette transition n'est pas suffisamment contrastée, ou si la sélection effectuée est vraiment trop approximative, un risque de débordement de la sélection voulue sera rencontré, cet incident correspond par exemple à la zone hachurée de la figure 13.

2.1.2.2 Propagation du contour

Plusieurs possibilités de propagation sont données à l'utilisateur : une méthode 2D de sélection avec propagation unidirectionnelle et une méthode 3D de sélection avec propagation sur un objet. Dans les deux cas il est possible d'effectuer une recherche en temps

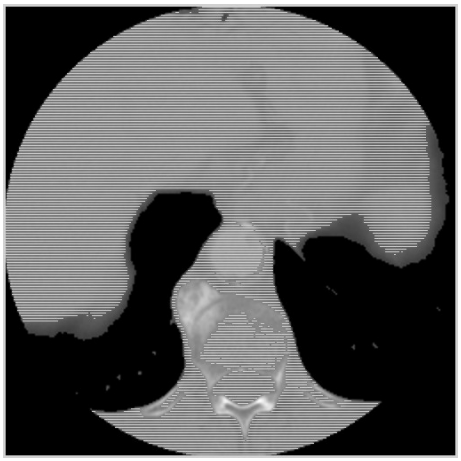
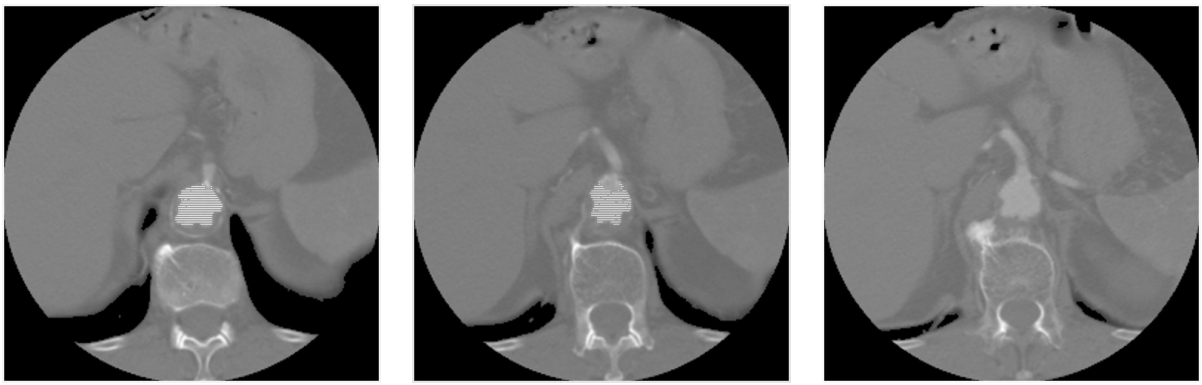


FIG. 13 – *Problème de sélection avec recherche en temps réel du contour le plus proche.*

réel du contour de la sélection. Ce dernier point a été décrit section 2.1.2.1.

2.1.2.3 Analyse de la méthode

La méthode 2D de sélection est la moins performante. Pour la détection de contour de l'aorte, cette méthode est inefficace. En effet, comme illustré figures 14.a, b et c, après une poursuite sur trois ou quatre images de la séquence, la sélection visible sur ces figures grâce aux zones hachurées, est complètement perdue et la détection s'arrête.

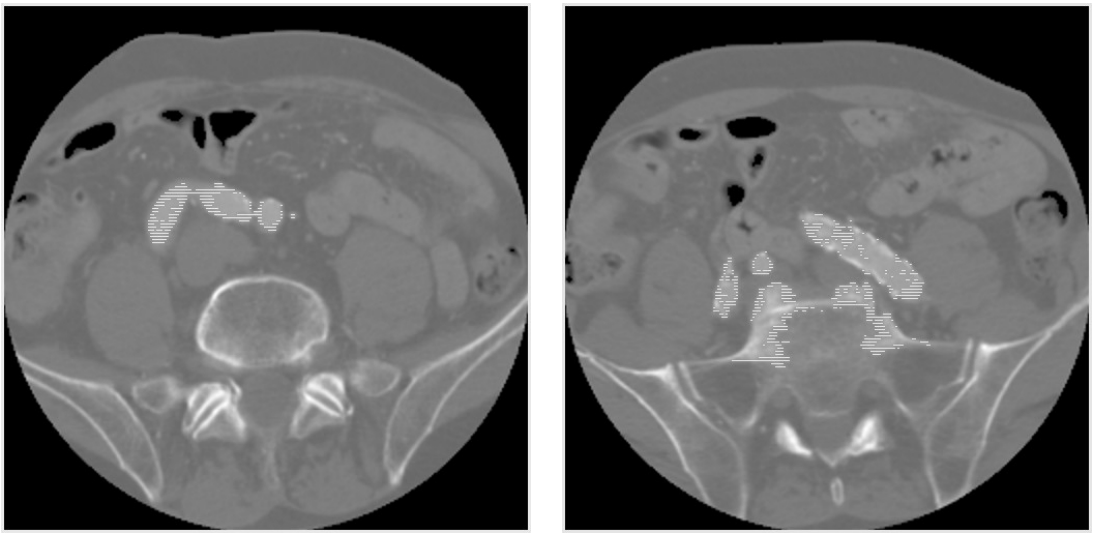


(a) (b) (c)

FIG. 14 – *a, b, c) Séquence de poursuite de l'aorte avec la méthode 2D de sélection. Perte de la zone de sélection après 3 images de la série.*

La méthode 3D de sélection permet parfois d'effectuer une poursuite complète de l'aorte sur toutes les images de la série. Si l'aorte ne contient pas de niveaux de gris trop variés, la poursuite pourra aboutir, et permettre une reconstruction 3D de la zone anatomique

ciblée, [Fiebich 97]. La méthode de poursuite s'effectue de manière très simple. C'est une détection de contour par seuil, [Cocquerez 95]. En effet, le contour à poursuivre sur l'image suivante se situe approximativement dans la même zone de l'image et contenant les mêmes niveaux de gris à un seuil près. Ce seuil détermine la précision de la poursuite. Plus ce seuil est petit, plus le risque de perte d'information sur l'aorte est grand, ceci se vérifie aisément sur la figure 15.a. La zone hachurée correspond au résultat de la détection de l'aorte. Plus ce seuil est important, plus le risque de débordement de la zone de sélection est grand, comme le montre la figure 15.b.



(a) (b)

FIG. 15 – *a) Perte d'information, méthode de seuil 3D avec sélection, b) Perte d'information sur les vaisseaux et débordement sur la colonne vertébrale.*

2.1.3 Amélioration par contours actifs de la méthode semi-automatique de détection d'une aorte

2.1.3.1 Détection des contours dans une coupe

La segmentation consiste à effectuer une partition de l'image en régions homogènes par rapport à un ou plusieurs critères. L'étape de prétraitement a pour but de faciliter la segmentation en renforçant la ressemblance entre pixels appartenant à une même région, ou en accentuant la dissemblance entre pixels appartenant à des régions différentes. Des méthodes de prétraitement peuvent être appliquées et concernent la modification d'histogramme, la réduction de bruit et le rehaussement de contraste.

Cette étape de prétraitement se justifie car les séquences d'images TDM sont bruitées

et peu contrastées dans les zones d'intérêts de notre étude. L'amélioration de la méthode présentée dans cette section repose sur l'utilisation de techniques de segmentation par contours actifs, [Tanguy 97, Smyth 96, Barbaresco 97]. L'initialisation du contour sera faite par le manipulateur en choisissant des points de contrôle [Cocquerez 95]. L'intérêt des contours actifs se justifie dans cette initialisation, [Ducottet 97, Kass 88]. L'algorithme de la bulle est aussi une technique possible pour initialiser le contour sur l'image. Un contour actif (ou snake) est une courbe se déformant après chaque itération. La courbe C est représentée selon les notations suivantes :

$$C = v(s,t) = [x(s,t),y(s,t)], \tag{3}$$

où s l'abscisse curviligne le long du contour avec $s \in [a,b]$, t avec a et b désignant les extrémités du contour, la variable temporelle $t \in [0,T]$ et $v(s,t)$ le point courant.

Le modèle est basé sur l'équation suivante :

$$E = \alpha E_{image} + E_{curv}, \tag{4}$$

où E représente l'énergie globale, E_{image} le terme énergétique lié à l'information image et E_{curv} le terme énergétique traduisant une hypothèse *a priori*. Cette hypothèse a priori faite sur le modèle concerne une contrainte de lissage spatial. Cette énergie E est définie localement, l'énergie totale résultant ensuite de la somme des énergies locales.

L'énergie image E_{image} fait intervenir les caractéristiques image que l'on cherche à mettre en valeur. Dans le cas précis où l'on cherche à mettre en valeur les zones de fort contraste, on peut choisir une énergie image donnée par la relation :

$$E_{image}(i,j) = g_h(i,j)^2 + g_v(i,j)^2, \tag{5}$$

où :

$$g_h(i,j) = \sum_{k=-1}^{+1} \sum_{l=-1}^{+1} SobelH(i,j) \times D_t(i+k,j+l), \tag{6}$$

et :

$$g_v(i,j) = \sum_{k=-1}^{+1} \sum_{l=-1}^{+1} SobelV(i,j) \times D_t(i+k,j+l), \tag{7}$$

avec $SobelH(i,j)$ et $SobelV(i,j)$ étant les masques dérivatifs.

Le terme E_{curv} exprime la contrainte de lissage faite sur la courbe C. On tente ainsi de réduire les angles trop aigus. L'expression de cette énergie suit l'équation suivante :

$$E_{curv} = angle(v(s-1),v(s),v(s+1))^2, \tag{8}$$

où $v(s)$ est le pixel de coordonnées s appartenant à C.

2.1.3.2 Suivi des contours dans les coupes

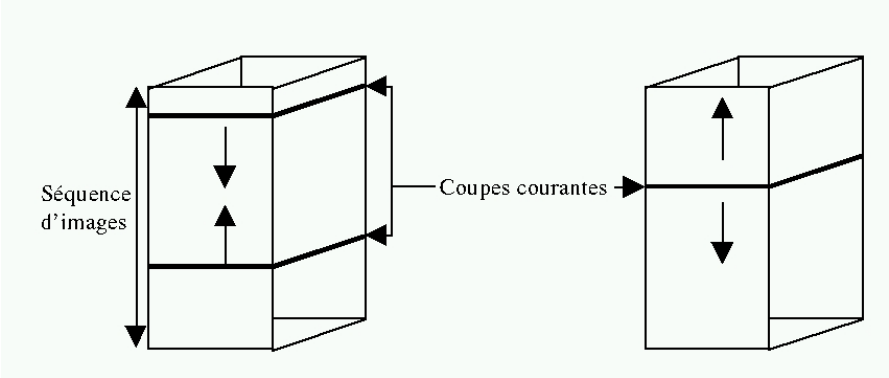


FIG. 16 – *Possibilité sur l'ordre de parcours des coupes.*

Le contour de la coupe précédente sert à initialiser le contour actif de la coupe courante. En effet, seules de légères déformations du contour se retrouvent entre deux coupes successives. Des travaux concernant de petites variations entre deux images d'une séquence ont déjà été développés pour des applications précises, [Latombe 97]. Nous proposons trois solutions concernant l'ordre de parcours des coupes :

- Comme représenté figure 16, il est possible d'initialiser deux contours sur les extrémités de la séquence d'images. Dans ce cas, la propagation des contours se fait vers la coupe centrale. Il est alors possible de comparer sur cette coupe les résultats obtenus par initialisations supérieure et inférieure.

- Une autre solution consiste à initialiser le contour sur l'image centrale et de le propager dans les deux sens de direction comme illustré figure 16.

- En utilisant un parcours séquentiel, il est possible de prendre trois coupes initiales équitablement réparties sur la séquence d'images en TDM. Si la séquence comporte n images, il convient de prendre trois coupes courantes, ($n/4$, $n/2$ et $3n/4$). A partir de chacune de ces coupes on lance une recherche de contours actifs dans les deux sens. Cette méthode permet un gain de temps de calcul considérable, comme expliqué figure 17.

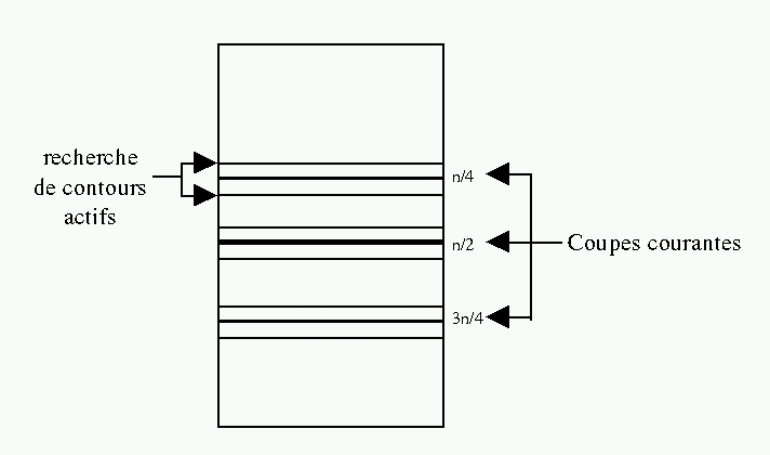


FIG. 17 – *Utilisation d'un parcours séquentiel.*

2.2 Propagation automatique de contours actifs dans une séquence d'images médicales

Dans cette partie nous présentons un nouvel algorithme permettant de suivre un organe dans une séquence d'images médicales afin de réaliser une reconstruction 3D. La méthode automatique que nous proposons permet de suivre le contour externe d'un organe anatomique dans toute la séquence à partir d'un contour initialisé par l'utilisateur sur la première image. Les opérations nécessaires pour notre méthode de suivi s'appuient sur une segmentation par contours actifs basée région. La localisation des objets avec une prédiction dynamique de déplacement est basée sur les fonctions de courbes de niveaux et sur la définition de région d'intérêt pour l'estimation locale robuste du modèle de l'image. Une application de cette méthode est la reconstruction 3D de l'aorte abdominale [Djemal 02, Djemal 05].

2.2.1 Introduction

Une solution possible pour reconstruire un objet 3D est de partir d'un ensemble d'images 2D [Berthilsson 97]. Notre travail s'insère dans cette direction. La méthode que nous avons développée est centrée sur l'algorithme de suivi, qui est la première étape de la reconstruction 3D. L'approche que nous proposons est basée sur plusieurs algorithmes. L'initialisation de l'algorithme de suivi est effectuée uniquement sur la première image de la séquence par un contour initial positionné par l'utilisateur. Un ensemble de contours obtenus est alors

utilisé pour le modèle et la reconstruction 3D. Notre méthode est basée sur la segmentation par contours actifs et sur la localisation automatique d'objets dans une séquence d'images médicales. Cet algorithme permet le suivi d'un objet changeant de forme et de topologie.
La forme de l'aorte abdominale dans une séquence d'images tomographique par rayons X peut changer en se rétrécissant ou en se dilatant. La topologie de l'objet peut aussi changer en se divisant ou en se fusionnant. De plus, l'image peut contenir d'autres types d'organes qui peuvent aussi changer de forme et de topologie.

Un anévrisme de l'aorte abdominale (AAA) est une dilatation de l'aorte abdominale. Si rien n'est fait un AAA continue de se dilater jusqu'à la rupture, qui est souvent suivie de la mort du patient. Dans le monde, 100 000 interventions chirurgicales pour la suppression d'un AAA sont effectuées tous les ans dont 30% sont endovasculaire. Beaucoup d'attention est nécessaire dans le suivi de patients après un traitement d'anévrisme endovasculaire [Fillinger 99].

Ce type de difficultés nous a motivé dans le fait d'exclure l'idée de prendre une image de référence dans notre stratégie de suivi. Dans les travaux de [Cotes 94, Kervrann 94], la méthode de segmentation de l'image est définie à partir d'une forme paramétrique obtenue par une base d'apprentissage. D'autres travaux [Fiebich 97, Jayaraman 97, Puech 00] utilisent des contours actifs géodésiques nécessitant une forme connue *a priori*. La propagation n'est pas automatique dans toutes les images de la séquence, et ces méthodes ne gèrent pas le changement de topologie. Une particularité de notre méthode est qu'elle suit le contour externe de l'aorte abdominale dans une séquence d'images. Dans ce but, nous définissons une région d'intérêt (RI) pour chaque image de la séquence. Nous développons également une estimation robuste du modèle de l'image. Sur la première image, cette RI est déterminée par une estimation globale. Sur la seconde image, le modèle est obtenu par estimation robuste locale dans un RI déterminée par dilatation du contour précédent. Finalement, pour les autres images de la séquence, la RI est obtenue par une prédiction dynamique de déplacement entre les deux précédents contours. Cette prédiction dynamique est obtenue par utilisation de fonctions de courbe de niveaux. L'établissement de cette méthode par courbe de niveaux autorise le changement de topologie et la localisation automatique d'objets.

Dans une première partie, nous exposons les méthodes existantes de contours actifs en présentant les méthodes basées régions et les méthodes basées contours. Nous montrons que les méthodes basées contours présentent des difficultés pour des images faiblement

contrastées et des contraintes d'initialisation. Ensuite, nous rappelons les méthodes de segmentation basées région qui sont utilisées dans le contexte de ce travail.

Dans le paragraphe suivant, nous présentons ensuite notre méthode de propagation automatique dans une séquence d'images médicales. Nous expliquons alors l'algorithme de segmentation et nos améliorations, ensuite, nous développons l'algorithme de localisation d'objet avec une estimation robuste locale. Finalement, dans le paragraphe résultat, nous appliquons notre méthode sur une séquence d'images médicales pour détecter les contours de l'aorte abdominale.

2.2.2 Contours actifs et contexte de travail

Depuis les travaux de [Kass 88] il y a 15 ans, deux classes de contours actifs ont été développées pour la segmentation des objets : les approches basées contours et les approches basées régions. La première approche utilise l'information située strictement le long des frontières. Les contours actifs évoluent vers les zones ayant le plus fort gradient d'intensité. Ces méthodes nécessitent une bonne initialisation. Les approches basées régions sont des outils puissants pour la segmentation, où l'information basée région doit être incorporée dans l'équation d'évolution du contour actif. Les snakes ont été introduits par [Kass 88] comme un modèle de contour actif pour la segmentation en région. Le modèle est dérivé du principe variationnel d'une mesure non géométrique. Le modèle est initialisé par une fonction d'énergie qui inclue les termes externes et internes qui sont intégrés le long d'une courbe. Soit la courbe $C(p) = (x(p),y(p))$, où $p \in [0,1]$ est une paramétrisation arbitraire. Le modèle de snake est défini par la fonction d'énergie :

$$E(C) \quad = \quad \alpha \int_0^1 (|C_p|^2)dp + \beta \int_0^1 (|C_{pp}|^2)dp - \gamma \int_0^1 g(C)dp, \tag{9}$$

où $C_p = (\partial_p x(p),\partial_p y(p))$ et α, β, γ sont des constantes positives.

Le dernier terme représente une énergie externe, où $g()$ est une fonction indicatrice de contours positifs qui dépend de l'image $f(x,y)$. Ce terme fournit de petites valeurs le long des frontières et de grandes valeurs ailleurs. On utilise par exemple $g(x,y) = \frac{1}{|\nabla f|^2+1}$. Le minimum d'énergie E est obtenu avec la courbe C, qui minimise E, solution d'une équation aux dérivées partielles (EDP) [Kass 88]. Le modèle de snake est un modèle linéaire et est donc un outil puissant et efficace pour la segmentation d'objets et l'intégration de contours, particulièrement quand il y a une approximation des frontières à détecter. Cependant le modèle n'est pas géométrique puisqu'il dépend de la paramétrisation. Les modèles de contours actifs géodésiques ont été introduits par [Caselles 95, Caselles 97] comme une

alternative géométrique des snakes. Le modèle est alors dérivé d'une fonction géométrique, où le paramètre arbitraire p est remplacé par une longueur d'arc Euclidien $ds = |C_p|dp$.

En effet, les modèles géométriques et géodésiques permettent une indépendance de la fonction d'énergie de la courbe de paramétrisation qui donne plus de stabilité au modèle de contour actif. Ainsi une convergence vers la solution la plus proche de la frontière des objets recherchés a été obtenue. De plus, des améliorations concernant la fonction d'arrêt des contours actifs ont été apportées. Même si ces fonctions sont principalement basées sur la norme du gradient de l'image originale, elles restent relativement puissantes si le gradient est très significatif. Mais ces solutions convergent difficilement vers une solution optimale si l'image est faiblement contrastée.

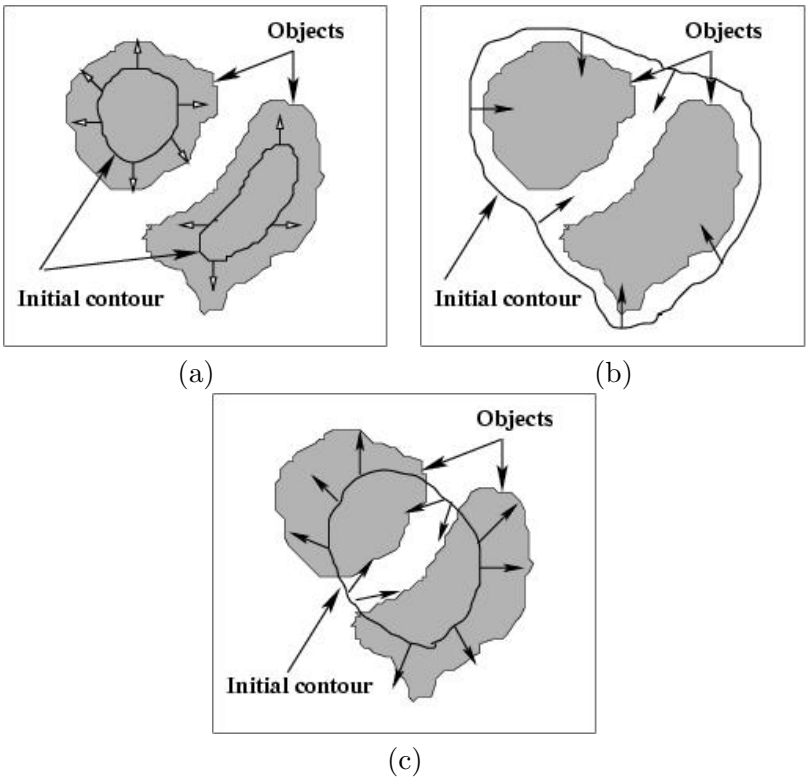


FIG. 18 – *Les différents cas d'initialisation a) Déplacement vers l'extérieur, b) Déplacement vers l'intérieur, c) Déplacement simultanément vers l'extérieur et l'intérieur.*

L'évolution de l'expression peut être seulement positive en des points déterminés, comme illustrée figure 18.a, ou négative en tous points comme illustrée figure 18.b. Le contour peut évoluer seulement dans une seule direction durant le traitement. Le cas illustré figure 18.c, ne peut pas être traité par ces méthodes car le contour doit évoluer simultanément dans les deux directions. Ce cas peut être rencontré dans une procédure

automatique de suivi d'objet dans une séquence d'images tel que le suivi de l'aorte abdominale par exemple. La forme de cet organe peut changer en se rétrécissant ou en se dilatant et la topologie peut se modifier en se divisant ou en fusionnant.

La méthode basée région consiste dans la définition d'un critère ou nous introduisons de l'information relative à chaque région de l'image. L'expression de la vitesse d'évolution est obtenu par la minimisation de ce critère. Des études récentes ont montré le potentiel de ces méthodes pour la segmentation des objets. Zhu et Yuille [Zhu 95, Zhu 96] ont présenté un cadre statistique pour la segmentation des images. Ils ont dérivé l'équation d'évolution du contour en minimisant un critère de Bayes (longueur de description minimum) généralisé inspiré de l'énergie de Mumford et Shah [Mumford 89]. Paragios et Deriche [Paragios 96] ont proposé une extension des travaux de Zhu et Yuille en changeant la fonction de descripteur de contour afin d'incorporer le gradient de l'image comme un contour actif géodésique [Caselles 97]. Dans leurs études, quelques fonctions de descripteurs ont été évaluées pour la segmentation de textures ou d'objets. Chakraborty *et al* [Chakraborty 96] ont introduit une approche basée région similaire pour la segmentation d'images médicales. Ce critère a été introduit et calculé numériquement afin de le maximiser. De la même manière, Chesnaud *et al* [Chesnaud 99] ont cherché à maximiser une fonction de vraisemblance en choisissant à chaque étape le meilleur déplacement du contour actif. Dans ces méthodes le contour est généré point par point. La gestion de changement de topologie est plus délicate.

D'autres travaux ont amélioré le modèle des contours actifs. Ainsi les formes complexes présentes dans les images peuvent être segmentées sans connaissance préalable de la topologie de l'objet. En effet, déplacé par la théorie de l'évolution des interfaces (courbes), [Caselles 93] et [Malladi 95] ont introduit les modèles géométriques qui prennent en compte les mesures géométriques internes et externes. A partir du modèle géométrique du contour actif [Sethian 96] ont introduit la formulation par courbe de niveau et ont montré que cette représentation implicite rendait possible la gestion automatique de changement de topologie.

Pour traiter le cas illustré figure 18.c, des difficultés existent dans le calcul de l'expression d'évolution permettant une évolution bidirectionnelle. Pour résoudre ce problème [Amadieu 99] ont proposé une méthode qui consiste dans la minimisation d'une critère variationnel d'un problème inverse. Ils ont dérivé le critère au sens des distributions pour obtenir une loi d'évolution du contour actif. Dans notre algorithme de suivi, la méthode de segmentation est basée sur les travaux de [Amadieu 99]. Cette méthode a l'avantage de

nécessiter qu'une seule initialisation de contour. Elle permet une formulation par courbe de niveau et gère les changements de topologie.

2.2.3 Propagation automatique dans une séquence d'images

Dans cette section, nous présentons notre méthode automatique de propagation de contours dans une séquence d'images. Dans le paragraphe 2.2.3.1, nous présentons l'algorithme de segmentation de contours actifs basés région et nos améliorations. A partir du contour obtenu dans la première image de la séquence, nous proposons de l'utiliser afin de détecter automatiquement tous les autres contours de la séquence. Dans le paragraphe 2.2.3.2, nous présentons l'algorithme de localisation avec prédiction dynamique de déplacement et la détermination de la RI pour une estimation robuste locale du modèle de l'image.

2.2.3.1 Modélisation de l'image et segmentation par contours actifs basée région

Comme annoncé section 2.2.2, la méthode de segmentation est basée sur les travaux de [Amadieu 99] à partir des études de [Santosa 96]. Dans cette section, nous rappelons le principe de la méthode et nous proposons des améliorations. Cette méthode nécessite l'utilisation des équations aux dérivées partielles (EDP). L'utilisation des EDP pour les contours actifs consiste dans le développement d'un contour C fonction de l'équation suivante :

$$\frac{\partial C}{\partial t} = F_c.\vec{N}, \tag{10}$$

où $\vec{N}$ est la normale à C, et F_c une expression d'évolution donnée dépendant de la segmentation du modèle de l'image.

Les contours actifs C évoluent perpendiculairement à eux mêmes, avec une expression d'évolution F_c jusqu'à ce qu'ils atteignent la frontière de l'objet à détecter. Le changement de topologie peut être obtenu en utilisant la formulation des courbes de niveaux [Sethian 96, Debreuve 99]. Dans cette méthode, la courbe C est définie comme le niveau zéro d'une surface déformable u. Si C évolue selon l'Eq. (10), alors la surface u évolue suivant cette EDP :

$$\frac{\partial u}{\partial t} = F_u.|\nabla u|. \tag{11}$$

Un changement topologique de C n'implique pas un changement topologique de u. Nous pouvons donc isoler un ou plusieurs objets pendant le même processus de détection. Le

modèle que nous avons choisi pour représenter l'image est :

$$f(x,y) = A(I(x,y)) + \eta(x,y), \tag{12}$$

où f est l'image originale, I le modèle, A un opérateur Gaussien et η un bruit. L'image est définie sur un domaine Ω avec :

$$I(x,y) \quad = \quad \left\{ \begin{array}{ll} I_1 & /(x,y) \in D_1 \\ I_2 & /(x,y) \in D_2 \end{array} \right. ; \left\{ \begin{array}{ll} D_1 & = \{(x,y)/u(x,y) < 0\} \\ D_2 & = \{(x,y)/u(x,y) > 0\}, \end{array} \right. \tag{13}$$

où :

$$D_1 \cup D_2 = \Omega, \text{et } (x,y) \in \Omega. \tag{14}$$

D_1 est le domaine de l'objet et D_2 le fond.

Le problème est de trouver le domaine D qui correspond au modèle I, où :

$$\partial D_t = C(t) = \{ \ (x,y) \ / \ u(x,y,t) = 0 \ \}. \tag{15}$$

La loi d'évolution de u est définie par la minimisation du critère :

$$J(t) \quad = \quad \sum_{i=1}^{p} \int_{\Omega} ||A(I_i(x,y,t)) - f(x,y)|| dxdy, \tag{16}$$

où $||\ ||$ est une norme choisie pour optimiser la séparation des différents objets et p le nombre de domaines. Dans notre cas nous avons $p = 2$.

A partir des études initiales de [Amadieu 99] et de nos expérimentations précédentes [Djemal 02], la diminution la plus rapide de $J(t)$ est obtenue pour l'expression d'évolution F_c sur $C(t)$:

$$F_c(x,y,t) = ||A(I_1(x,y)) - f(x,y)|| - ||A(I_2(x,y)) - f(x,y)||. \tag{17}$$

Comme nous pouvons voir dans l'Eq. (17), l'expression d'évolution peut être positive ou négative. Les contours actifs peuvent évoluer simultanément dans les deux directions, intérieure et extérieure. Par conséquent, la méthode devient plus flexible et les contraintes d'initialisation diminuent, comme illustré figure 18.c. L'EDP (10) devient :

$$\frac{\partial C}{\partial t} = \big(||A(I_1(x,y)) - f(x,y)|| - ||A(I_2(x,y)) - f(x,y)||\big)\vec{N}. \tag{18}$$

En prenant en compte la minimisation de la longueur du contour [Caselles 97], nous pouvons considérer le nouveau critère pour être minimisé de la manière suivante :

$$G(t) = \sum_{i=1}^{p} \int_{\Omega} ||A(I_i(x,y,t)) - f(x,y)|| dxdy + \lambda \int_{C(t)} ds, \tag{19}$$

où λ est un paramètre qui modifie l'influence de régularisation du contour et ds la longueur euclidienne de l'arc.
La minimisation de $G(t)$ est équivalente à minimiser à la fois $J(t)$ donné par l'Eq. (16) et le terme de régularisation de contours en utilisant la même fonction de courbes de niveaux. L'implémentation des courbes de niveaux garantit que les changements topologiques sont naturellement pris en compte. Cela permet la détection de tous les objets qui apparaissent dans le plan de l'image sans connaissance exacte de leur nombre. L'EDP (11) devient :

$$\frac{\partial u}{\partial t} + (F_c(x, y, t) + \lambda.\kappa(x, y, t))|\nabla u| = 0, \tag{20}$$

avec κ la courbure.

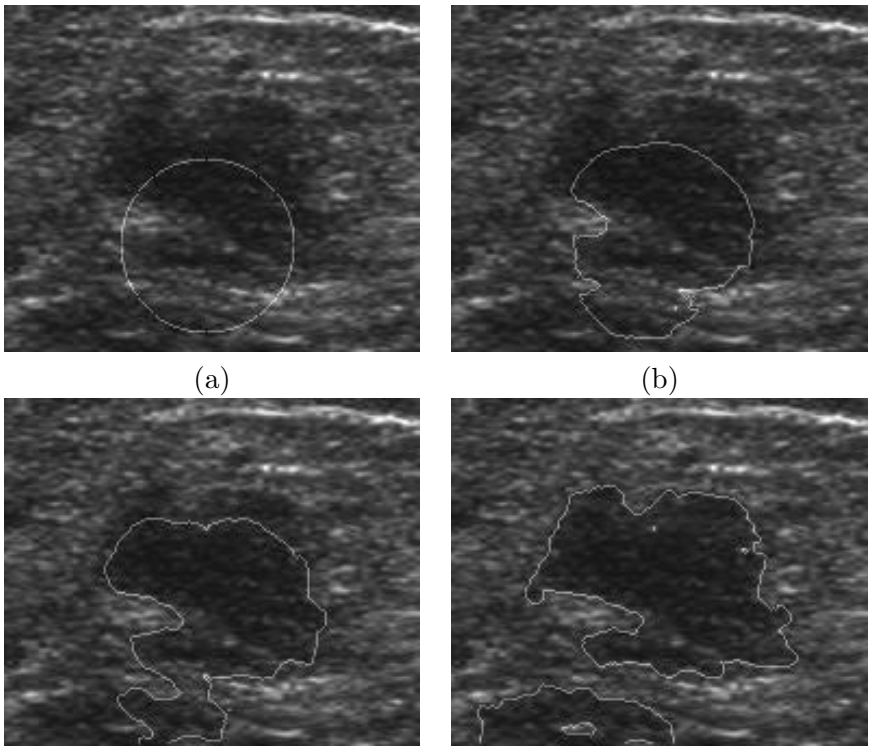


FIG. 19 – *Résultat de segmentation sur une image échographique d'une tumeur du sein avec un terme de régularisation optimal $\lambda = 100$: a) Initialisation b) et c) Propagation d) Convergence afin d'obtenir le contour final représentant la région de la tumeur.*

Comme nous l'avons montré précédemment, l'expression d'évolution obtenue par l'Eq. (17) dépend de l'image originale. Afin d'accélérer la convergence de l'algorithme, nous définissons une nouvelle expression d'évolution en l'affectant avec une fonction à coefficient multiplicatif de la norme du gradient de l'image originale f. La nouvelle EDP qui permet

l'évolution du contour est :

$$\frac{\partial u}{\partial t} + \Big(F_c(x, y, t).|\nabla f|^{\delta} + \lambda.\kappa(x, y, t)\Big)\,|\nabla u| = 0, \tag{21}$$

où $|\nabla f|$ est une évaluation numérique de la norme du gradient obtenue par convolution avec la dérivée d'une Gaussienne de l'image f et d'une constante positive δ.

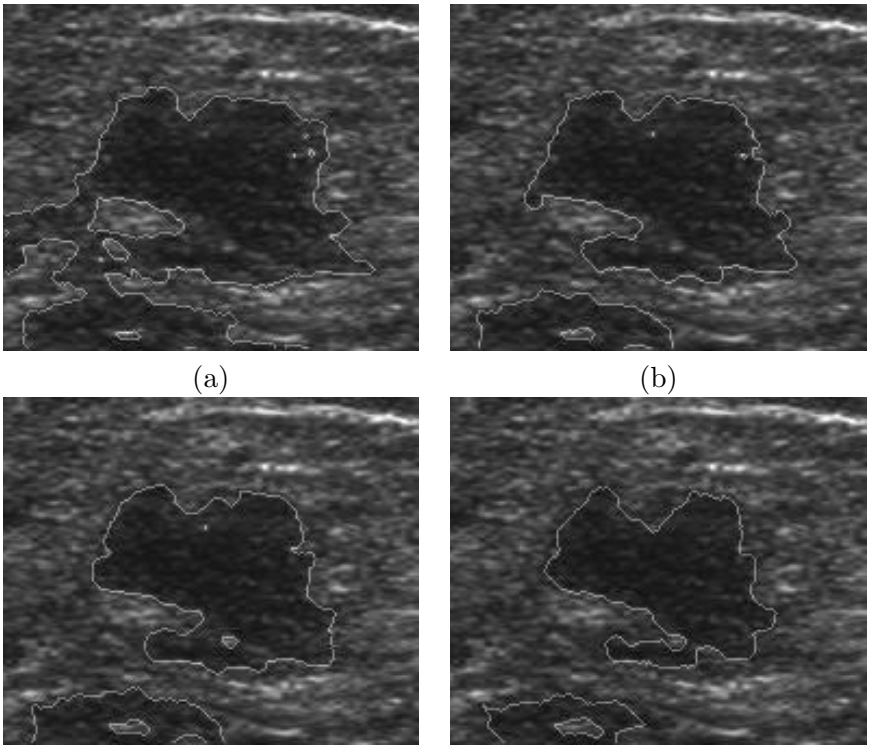


FIG. 20 – *Effet du terme de régularisation λ : a) $\lambda = 0$, convergence sans régularisation, b) $\lambda = 100$, convergence avec régularisation optimale, c) $\lambda = 200$ et d) $\lambda = 400$, lissage trop important des contours et perte de l'information.*

Sa valeur est donnée en accord avec l'image originale. Dans la figure 19, la propagation du contour est testée avec et sans le terme $|\Delta f|^{\delta}$ montré dans l'équation (21). La convergence avec le terme est obtenue après 273 itérations et sans le terme après 315 itérations. Le terme de régularisation λ a un effet sur le contour figure 20. Par exemple, avec une valeur nulle pour ce terme, figure 20.a, la propagation est faite sans régularisation. De même, pour de faibles valeurs de λ, nous segmentons des régions qui ne correspondent pas exactement aux objets. D'un autre coté, si on augmente trop cette valeur, le lissage est trop significatif, et donc des parties de contours à forte courbure sont mal traitées figures 20.c et d [Djemal 03b]. Par conséquent nous pouvons perdre de l'information. La

valeur optimale de λ, figure 20.b, peut être choisie par un docteur spécialisé en imagerie médicale. La formulation des courbes de niveaux permet une estimation effective des propriétés géométriques du contour C comme la courbure κ et le vecteur normal unitaire $\vec{N}$. Ces propriétés sont estimées par :

$$\kappa = div\left(\frac{\nabla u}{|\nabla u|}\right) \quad \text{and} \quad \vec{N} = -\frac{\nabla u}{|\nabla u|}. \tag{22}$$

Dans cette section, l'image $I(x, y)$ est supposée connue ou donnée. Cela rend possible une estimation préliminaire qui est implémentée par un estimateur robuste, présenté dans la section 2.2.3.4.

2.2.3.2 Région d'intérêt et estimation robuste locale

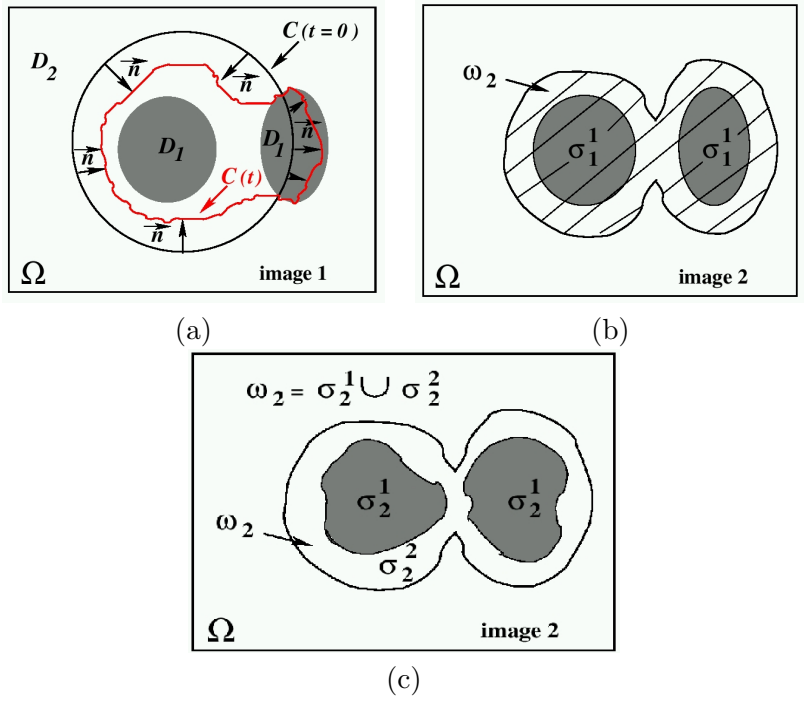


FIG. 21 – *a) Domaine de l'objet et le fond dans la première image, b) Dilatation de la région σ_1^1 de l'objet de la première image dans la seconde image, c) Propagation de la région dilatée à partir de σ_1^1 pour obtenir ω_2, la RI dans la seconde image.*

La principale hypothèse considérée dans cette section est que les variations entre deux coupes successives sont petites. Dans la section 2.2.3.1, nous considérons que l'image contient deux régions, D_1 les objets et D_2 le fond. L'estimation robuste du modèle de l'image sur la première coupe de la séquence est faite sur l'image complète. A partir du contour obtenu sur la première image, nous définissons une RI, appelée ω_2, pour la seconde

image. Cette RI contient l'objet désiré et nous permet d'appliquer une estimation robuste locale du modèle de l'image. Cette RI pour la seconde image est donnée par :

$$\omega_2 = \beta.\sigma_1^1 = \{(x_0, y_0) + \beta(x - x_0, y - y_0) \; \forall \; (x, y) \in \sigma_1^1\}, \tag{23}$$

où β est le facteur de dilatation avec $\beta > 1$ et (x_0, y_0) un point appartenant à σ_1^1. Cela correspond à la région de l'objet dans la première image où $\sigma_1^1 = D_1$ comme illustré figures 21.a et 21.b. Dans un espace de corps morphologiques, des opérateurs de base nous permettent de modifier un corps morphologique. Ces modifications sont reliées à des éléments structurants [Serra 88]. La dilatation et l'érosion sont une addition et une soustraction particulière d'un élément structurant d'un corps morphologique.

La région ω_2 peut également être considérée par :

$$\omega_2 = \sigma_2^1 \cup \sigma_2^2, \tag{24}$$

où σ_2^1 correspond à la région de l'objet et σ_2^2 correspond au fond seulement dans la RI ω_2 pour la seconde image.

La RI ω_2 est juste une partie du domaine de l'image Ω, comme illustrée figure 21.c.

Pour obtenir cette région, nous devons premièrement remplir la région limitée par le contour trouvé dans l'image précédente afin d'obtenir σ_1^1. Deuxièmement nous devons dilater cette région avec un facteur de dilatation β, comme montré figures 21.b et 21.c.

2.2.3.3 Prédiction dynamique des déplacements et localisation

Dans cette section, nous montrons comment une prédiction dynamique des déplacements pour localiser les objets est possible. Cette prédiction commence dès la troisième coupe. Si nous considérons seulement la dilatation des contours, comme nous avons déjà montré, plusieurs problèmes sont rencontrés. Ces problèmes consistent dans le choix d'une valeur du facteur de dilatation β adaptée à toutes les images de la séquence. En effet, si la dilatation est trop grande, l'estimation robuste locale sera moins précise et de plus la RI peut contenir une partie d'un objet non désiré.

D'un autre coté si la dilatation est trop petite, une partie de l'objet désiré peut ne pas être incluse dans la RI. Pour essayer d'éliminer ces problèmes, nous avons développé une méthode de localisation par prédiction dynamique des déplacements de l'objet dans toute la séquence. Cette méthode est basée sur les fonctions de courbes de niveaux. A partir de la troisième coupe de la séquence, nous appliquons une estimation du déplacement de l'objet entre les deux coupes précédentes. Cette estimation permet de mieux définir une

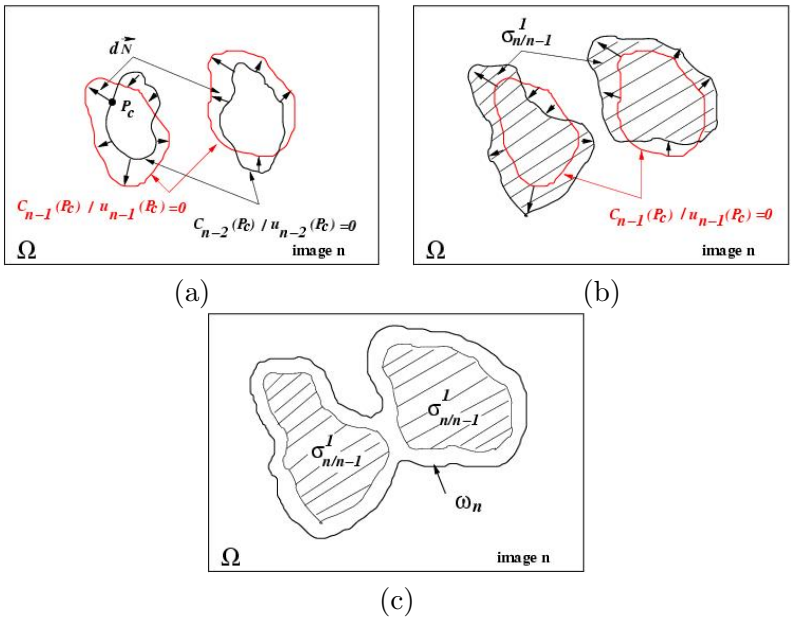


FIG. 22 – *a) Estimation du déplacement en les contours C_{n-2} et C_{n-1} tel que $(p/u_{n-2}(p) = 0)$ et $(p/u_{n-1}(p) = 0)$, b) Détermination de la région $\sigma^1_{n/n-1}$ en tenant compte du déplacement, c) Dilatation de la région $\sigma^1_{n/n-1}$ pour obtenir ω_n, la RI dans l'image n.*

RI dans laquelle nous estimons le modèle de l'image et nous permet de mieux contrôler la dilatation des contours.

Soit p_c un point de la frontière de l'objet dans l'image $n-2$. La déformation de l'objet (aorte abdominale) dans l'image $n-1$ donne un point appartenant à $C_{n-1}(p_c)$. Si nous notons $d(p_c)$, le déplacement de p_c entre les images $n-2$ et $n-1$, nous avons $C_{n-1}(p_c) = p_c + d(p_c)$, comme illustré figure 22. Les contours $C_{n-1}(p_c)$ et $C_{n-2}(p_c)$ représentent respectivement les niveaux zéro des courbes $u_{n-1}(p)$ et $u_{n-2}(p)$. Celles ci sont les fonctions obtenues à la convergence de l'algorithme de segmentation dans les deux images $n-2$ et $n-1$ telles que $\{p \ / \ u_{n-1}(p) = 0\}$ et $\{p \ / \ u_{n-2}(p) = 0\}$.

La déformation de l'objet de l'image $n-2$ à l'image $n-1$ permet de faire évoluer la fonction de courbes de niveaux u_{n-2} vers la fonction de courbes de niveaux u_{n-1}. Nous voulons estimer le déplacement de l'objet en utilisant ces fonctions. Nous considérons les points p_c de la frontière de l'objet tels que $u_{n-2}(p_c) = 0$. Nous supposons que $p_c + d(p_c)$ appartient à la frontière de l'objet dans l'image $n-1$. Nous obtenons donc l'équation suivante :

$$u_{n-1}(p_c) = u_{n-2}(p_c + d(p_c)). \tag{25}$$

Le vecteur de déplacement du point p_c est normal à la direction du contour $C_{n-2}(p_c)$. De

plus, ce déplacement est estimé dans le sens du vecteur normal unitaire, Eq. (22), comme illustré figure 22.a avec :

$$d(p_c).\vec{N} = [u_{n-1}(p_c) - u_{n-2}(p_c)].\frac{\nabla u_{n-2}(p_c)}{|\nabla u_{n-2}(p_c)|}. \tag{26}$$

Après l'estimation de déplacement, nous définissons une fonction de courbes de niveaux intermédiaires, appelée $u_{n/n-1}(p)$ où :

$$u_{n/n-1}(p_c) = u_{n-1}(p_c + d(p_c)). \tag{27}$$

La fonction de courbes de niveaux intermédiaires obtenue $u_{n/n-1}$ nécessite une interpolation des points du contour obtenu à partir de la fonction u_{n-1} après avoir pris en compte les déplacements. L'interpolation devient très difficile si l'objet suivi est divisé en plusieurs

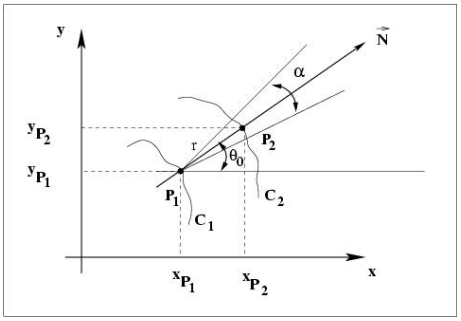


FIG. 23 – *Définition des arcs de cercle, centrés autour d'un point du contour.*

régions, et par conséquent en plusieurs contours. Pour cela, et pour donner une solution à cette interpolation, nous définissons des arcs de cercles de rayon $r = d(p_c)$ et d'angle α, illustré figure 23.

Nous considérons un cercle de centre (x_{p_1}, y_{p_1}) et de rayon r, qui peut être représenté par la paramétrisation suivante :

$$\begin{cases} x(\theta) = r.cos\ \theta + x_{p_1} \\ y(\theta) = r.sin\ \theta + y_{p_1} \end{cases} \quad , \quad \theta \in [0, 2\pi]. \tag{28}$$

Cet arc de cercle est alors noté A et centré autour du point p_2 (figure 23) et défini par :

$$A = \left\{ (x(\theta), y(\theta)) \ / \ \theta \in \left[\theta_0 - \frac{\alpha}{2} \ , \ \theta_0 + \frac{\alpha}{2} \right] \right\}, \tag{29}$$

où α est donné.

De cette manière, l'implémentation par courbes de niveaux nous permet de prendre en compte automatiquement plusieurs contours. La RI sans dilatation, montrée figure 22.b, est définie par :

$$\sigma^1_{n/n-1} = ((x,y)/u_{n/n-1}(x,y) \leq 0). \tag{30}$$

Donc la RI pour l'image n, illustrée figure 22.c, est obtenue par :

$$\omega_n = \beta.\sigma^1_{n/n-1} = \beta.((x,y)/u_{n/n-1}(x,y) \leq 0). \tag{31}$$

Dans cette région ω_n, par cette méthode nous améliorerons l'estimation robuste locale du modèle de l'image. De plus cela diminue le nombre d'opérations et améliore la localisation des objets désirés.

2.2.3.4 Estimation locale robuste

Dans la section 2.2.3.1, le modèle de l'image est représenté par les deux constantes I_1 et I_2 relatives aux domaines D_1 et D_2, qui sont respectivement l'objet recherché et le fond. Dans cette section, nous présentons l'estimation locale robuste du modèle de l'image. Les deux constantes I_1 et I_2 sont estimées sur la globalité de la première image avec le robuste estimateur [Odobez 95b, Odobez 95a]. Cet estimateur est formalisé par un problème des moindres carrés pondérés :

$$\hat{I} = Arg\ min \sum_i \frac{1}{2}\rho_i(f_i - I)^2, \tag{32}$$

avec :

$$\begin{cases} \rho_i = (1 - (\frac{|f_i - I|}{c})^2)^2 & /|f_i - I| \leq c \\ \rho_i = 0, & /|f_i - I| > c, \end{cases} \tag{33}$$

où c représente la valeur maximale du résidu pour limiter la contribution de certains points et $i \in [1,N]$, si N est le nombre de pixels dans l'image.

Nous avons noté que la variation d'intensité lumineuse de l'image entre deux coupes voisines peut changer aléatoirement. Notons que cette estimation globale nécessite plus de temps de calcul. De plus, il est nécessaire de supprimer manuellement les autres objets contenus dans le domaine global Ω de l'image, comme par exemple la colonne vertébrale. De plus, la valeur du paramètre c doit être reconsidéré pour chaque image de la séquence.

Dans le cas d'une estimation locale robuste avec ω_n, la valeur de c est calculée une seule fois à partir de la seconde image. Nous pouvons alors utiliser cette valeur pour toutes les images de la séquence.

L'estimation locale robuste est basée seulement sur la RI ω_n. Dans cette région, la contribution des autres objets est quasi nulle. Cette contribution améliore la qualité de l'estimation de l'objet et diminue le temps de calcul comme illustré Tableau 1, où le temps d'estimation peut être considéré comme inversement proportionnel à σ_1/ω_n. A partir de

l'Eq. (32) l'estimation locale robuste de l'objet est :

$$\hat{I}_1 = \frac{1}{\sum_i \rho_i} \sum_i \rho_i I_1. \tag{34}$$

Dans le cas d'une estimation locale robuste, la stabilité de la valeur de c montre que le fond σ_n^2 est une partie minoritaire de la région locale ω_n. En fait, à partir de la coupe précédente, nous pouvons estimer approximativement la nouvelle région σ_n^1 de l'objet. Par conséquence, l'estimation locale robuste est basée principalement sur la région de l'objet recherché, car le nombre de point appartenant à σ_n^1 est supérieur au nombre de points de σ_n^2.

2.2.3.5 Résultats expérimentaux

Dans cette section, nous appliquons nos méthodes à une partie d'une séquence d'images médicales. Nous utilisons sept coupes (sous-séquence) du centre de cette séquence. Dans la section 2.2.3.6, nous présentons le résultat de la propagation du contour actif dans la première image. Nous illustrons sur cette sous-séquence la méthode de contours actifs et la propagation entre la première et la seconde image, section 2.2.3.7. Dans la section 2.2.3.8, nous présentons les résultats de la propagation dynamique sur les autres images de la sous-séquence.

2.2.3.6 Calcul du contour actif sur la première image de la séquence

Dans la figure 24, nous montrons le résultat de la segmentation obtenue sur la première image de la séquence de sept coupes représentant une partie de l'aorte abdominale.

L'évolution est montrée étape par étape en commençant par la forme initiale sélectionnée $u(x, y, t = 0)$ dont le niveau zéro de la courbe représente le contour initial $C(t = 0)$ (figure 24.a). Progressivement, avec l'évolution de l'algorithme, la fonction de courbe de niveaux $u(x, y, t)$ est déformée afin de représenter le contour en prenant en compte les changements de topologie, figures 24.b et 24.c. La fonction progresse jusqu'à se stabiliser quand le minimum du critère (19) est atteint. Alors la courbe de niveau zéro représente le contour final des objets (figure 24.d). La convergence est obtenue après 356 itérations avec cette image de taille 256×256 pixels. Le nombre d'itérations diminue sur les autres images de la séquence car l'initialisation est faite par le contour obtenu sur la précédente image. Pour chaque application de notre algorithme de segmentation, les valeurs des paramètres à avoir sont l'estimation robuste du paramètre c (décrit section 2.2.3.4) et le terme de

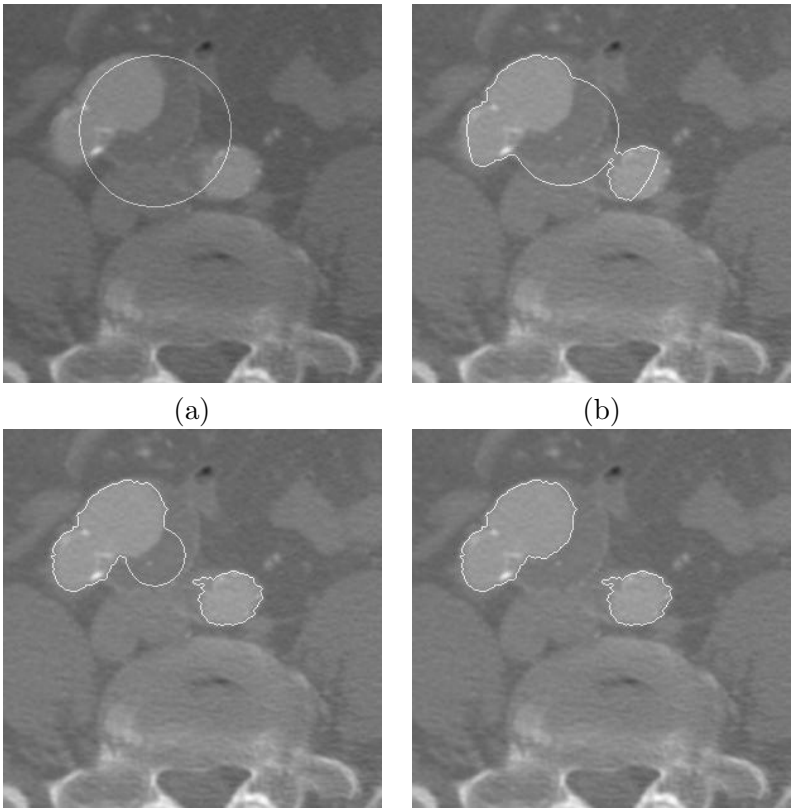


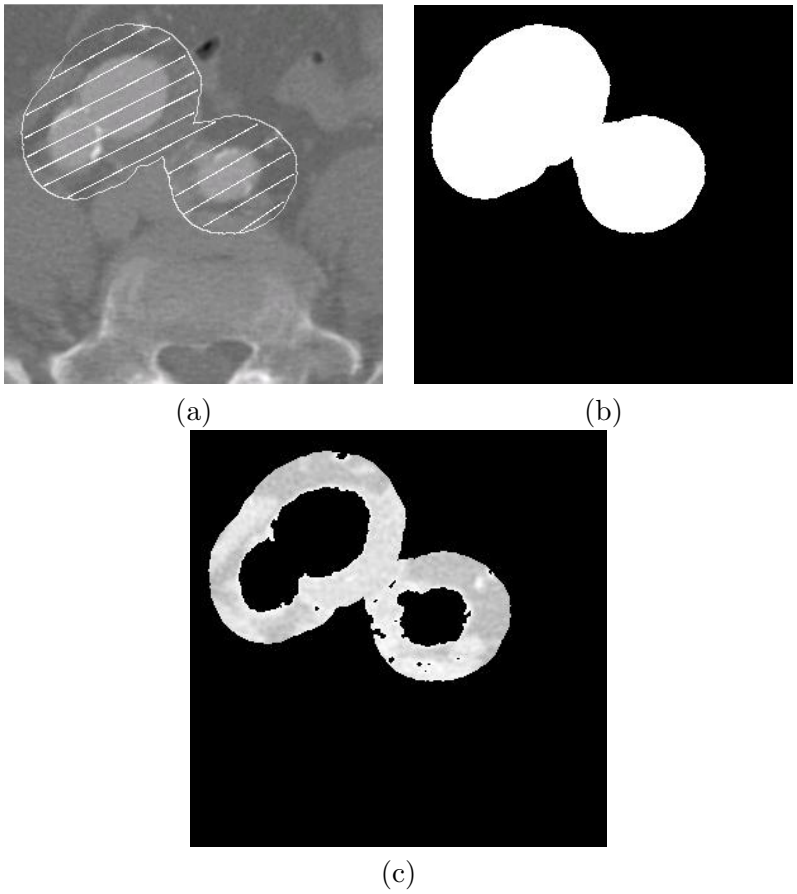
FIG. 24 – *Processus de segmentation relatif à la première coupe de la séquence : a) Initialisation, b) et c) Différentes étapes de la propagation, d) Convergence.*

régularisation λ. Pour cette première image, la valeur optimale du terme de régularisation est $\lambda = 150$ avec un pas de $\Delta t = 0.0001$ seconde. A partir seulement d'un cercle qui coupe l'objet, l'expression d'évolution peut être positive en certains points et négative en d'autres points. Par conséquent, le contour peut se propager dans les deux directions.

L'algorithme de segmentation est :

Initialisation : λ, c, Δt, $u^{i,j}_{(t=0)}$
Estimation_robuste : $\Rightarrow$ modèle de l'image I_1 et I_2
Calcul : expression d'évolution $F^{i,j}_c$
Tant que la convergence n'est pas atteinte
    Calcul : courbure $\kappa^{i,j}_t$ et $\left|\nabla^{i,j} u^{i,j}_t\right|$
    Calcul : $u^{i,j}_{(t+1)}$
    Extraction du contour : $u^{i,j}_{(t+1)} = 0$
    Reconstruction : $u^{i,j}_{(t+1)}$

2.2.3.7 Région d'intérêt, estimation locale robuste et propagation du contour actif pour la seconde image de la séquence



(a) (b)

(c)

FIG. 25 – *a) A partir du contour obtenu sur la première coupe, figure 24.d, nous obtenons la RI ω_2, b) Remplissage de la RI ω_2 pour estimer le modèle de l'image, c) Résultat de l'estimation locale robuste de l'aorte pour la seconde coupe de la séquence.*

A partir du contour obtenu sur la première image, figure 24.d, nous pouvons appliquer une dilatation de ce contour, figure 25.a, pour couvrir complètement la surface de l'objet dans la coupe suivante, figure 25.b. L'estimation locale robuste peut être faite dans cette RI ω_2 comme illustrée figure 25.c.

Le paramètre utilisé c est le seuil qui nous permet de démarrer le processus. Expérimentalement pour l'estimation globale, la valeur de ce seuil c est optimale pour $1 < c < 8$, cette valeur peut changer pour chaque image. Dans le cas de notre expérience d'estimation locale robuste, nous avons une valeur optimale $c = 5$ pour toutes les images de la séquence. Nous initialisons le contour sur la seconde image avec le contour obtenu sur la première image, figure 26.a. Après la propagation, figures 26.b et 26.c, l'algorithme converge dans

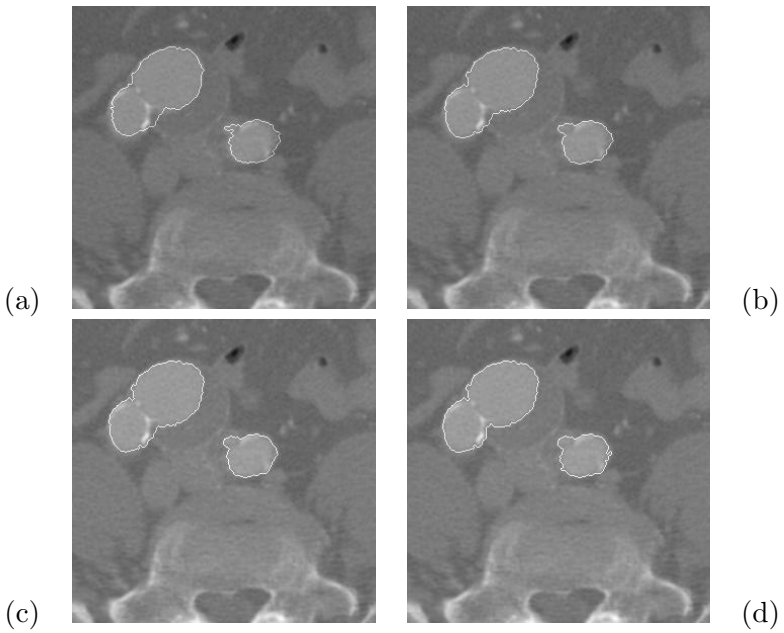


FIG. 26 – *Processus de segmentation relatif à la seconde image de la séquence, a) Initialisation avec le contour de la première image, b) et c) Différentes étapes de la propagation, d) Convergence afin d'obtenir le contour final.*

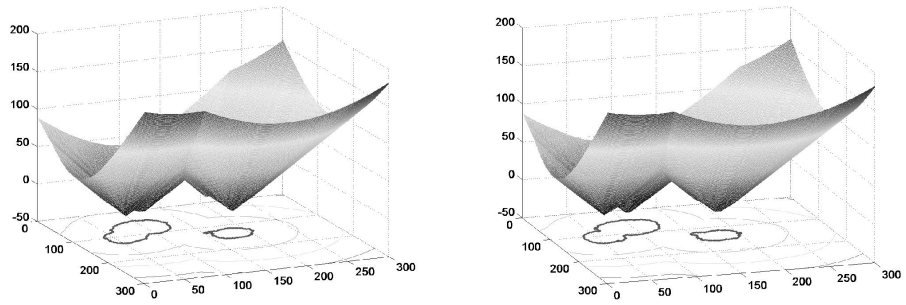
la seconde image pour obtenir le contour montré en figure 26.d.

2.2.3.8 Propagation dynamique des contours actifs dans les autres images de la sous-séquence

Les figures 27.a et 27.b illustrent la fonction de courbes de niveaux u_1 et u_2 respectivement pour la première et la seconde images de la sous-séquence. Le niveau zéro de ces fonctions correspond aux contours des figures 24.d et 26.d.

Pour la troisième image de notre sous-séquence, nous estimons le déplacement entre les deux premières images pour localiser l'objet, figure 28.a. La figure 28.b illustre la dilatation pour obtenir ω_n.

Après remplissage de la RI figure 28.c, nous pouvons obtenir l'estimation locale robuste de l'aorte dans la troisième coupe, Figure 28.d. Nous effectuons donc l'initialisation dans la troisième image avec le contour final obtenu dans la seconde image, figure 29.a, afin d'obtenir le contour final après convergence comme montré figure 29.b. Les figures 31.a illustrent la RI obtenue après prédiction dynamique des déplacements et dilatations. Les figures 31.c montrent les résultats des contours obtenus après initialisation avec les contours précédents, illustrés figures 31.b, sur les quatrième, cinquième, sixième et septième coupes de la séquence.

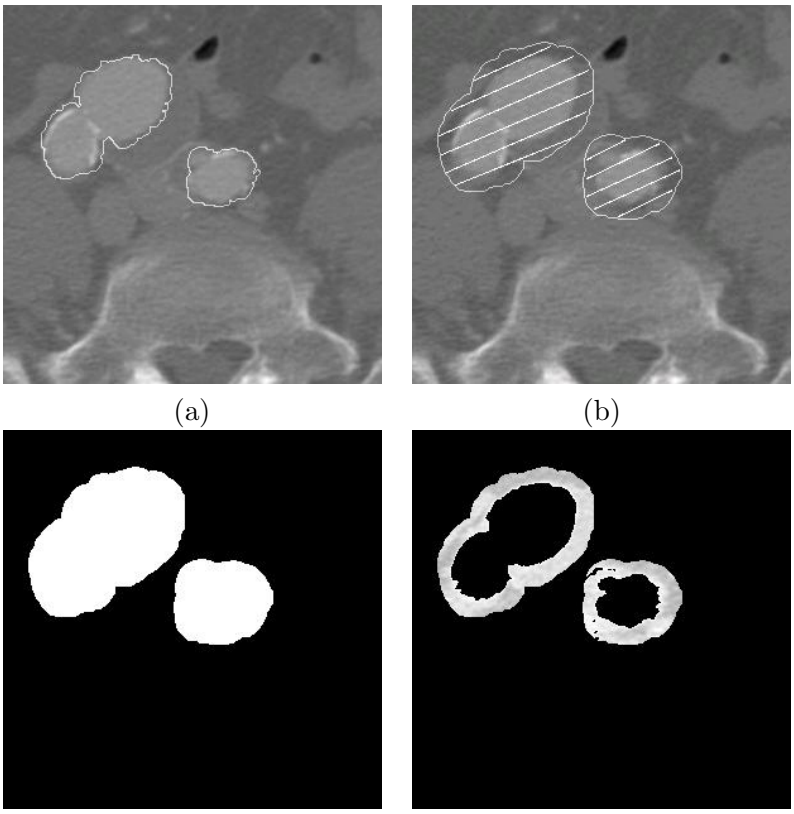


(a) (b)

FIG. 27 – *Les fonctions de courbes de niveaux : a) La fonction de courbe de niveaux u_1 après convergence dans la coupe* 1*, b) la fonction de courbe de niveau u_2 après convergence dans la coupe* 2*.*

TAB. 1 – *Nombre de pixels dans l'objet et la région du fond région pour chaque image de la sous-séquence.*

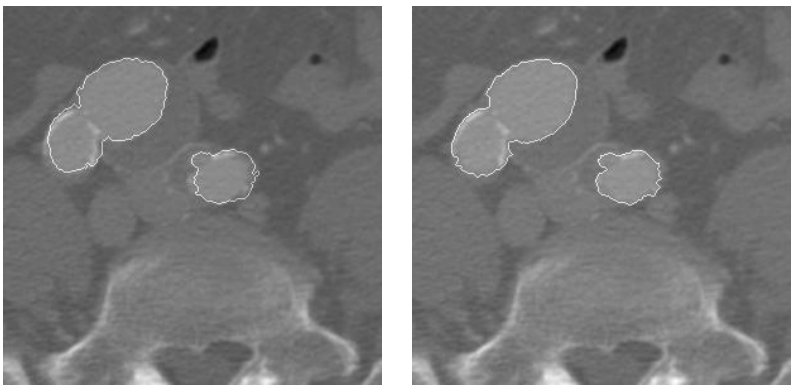
coupes	estimation	nombre total de pixels ω_n	pixels du fond σ_n^2	pixels de l'objet σ_n^1	rapport % $\frac{\sigma_n^1}{\omega_n}$
coupe 1	entière	65536	59006	6530	9.96
coupe 2	locale	21454	14536	6918	32.25
coupe 3		14775	7465	7010	48.43
coupe 4	locale	15761	9176	6585	41.78
coupe 5	avec	15158	8612	6546	43.19
coupe 6	prédiction	15850	7674	8176	51.59
coupe 7	(dynamique)	15493	8659	6834	44.11



(a) (b)

(c) (d)

FIG. 28 – *a) Résultat de la prédiction dynamique de déplacement et de localisation obtenu à partir des deux coupes précédentes donnant $\sigma^1_{n/n-1}$ dans la troisième coupe, b) Dilatation avec β pour obtenir ROI dans la troisième coupe, c) Remplissage de la troisième coupe de la ROI afin d'estimer le modèle de l'image, d) Résultat de l'estimation robuste locale de l'aorte dans la troisième coupe.*



(a) (b)

FIG. 29 – *Processus de segmentation pour la troisième coupe de la séquence, a) Initialisation, sur la troisième coupe, avec le contour obtenu sur la seconde coupe figure 26.d b) Convergence afin d'obtenir le contour final dans la troisième coupe.*

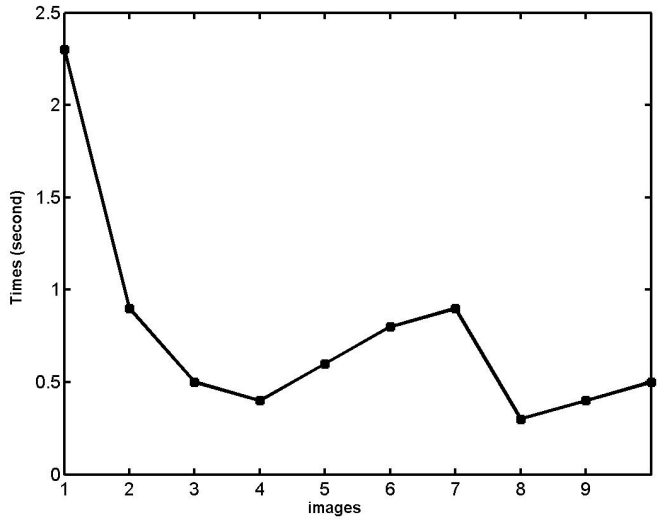


FIG. 30 – *Temps de calcul pour chaque image pour obtenir le contour final.*

Dans le Tableau 1, nous avons analysé pour chaque coupe de la sous-séquence le nombre de pixels contenus dans les régions de l'objet et du fond. Nous constatons que l'estimation locale robuste améliore la qualité de l'estimation. De plus, de la troisième à la septième coupe, avec la prédiction dynamique, la région de l'objet σ_n^1 correspond environ à la moitié de la région d'estimation ω_n. Nous pouvons également observer, figure 30, que le temps de calcul est plus important pour la première et la seconde coupe que pour les autres images de la séquence. A partir de la troisième séquence, le temps de calcul dépend du déplacement de l'objet entre deux coupes

Avec les contours finaux obtenus à partir des sept coupes de la séquence, comme illustrés figures 24.d, 26.d, 29.b et 31.c, nous pouvons développer une reconstruction 3D. Nous pouvons voir, figures 32, trois vues de cette reconstruction 3D. Nous avons également appliqué notre méthode de reconstruction 3D de l'aorte sur une séquence de 69 coupes. Le résultat obtenu à partir des contours finaux de chacune des 69 coupes est présenté figures 32.

L'algorithme de localisation est:

Fonctions des courbes de niveaux obtenues : à partir des images $n-2$ et $n-1$
Calculer : déplacements entre la courbe de niveau zéro de u_{n-2} et u_{n-1}
Calculer : u_{n-1} + déplacements
Construire avec interpolation : $\Rightarrow u_{n/n-1} \Rightarrow \sigma_{n/n-1}$

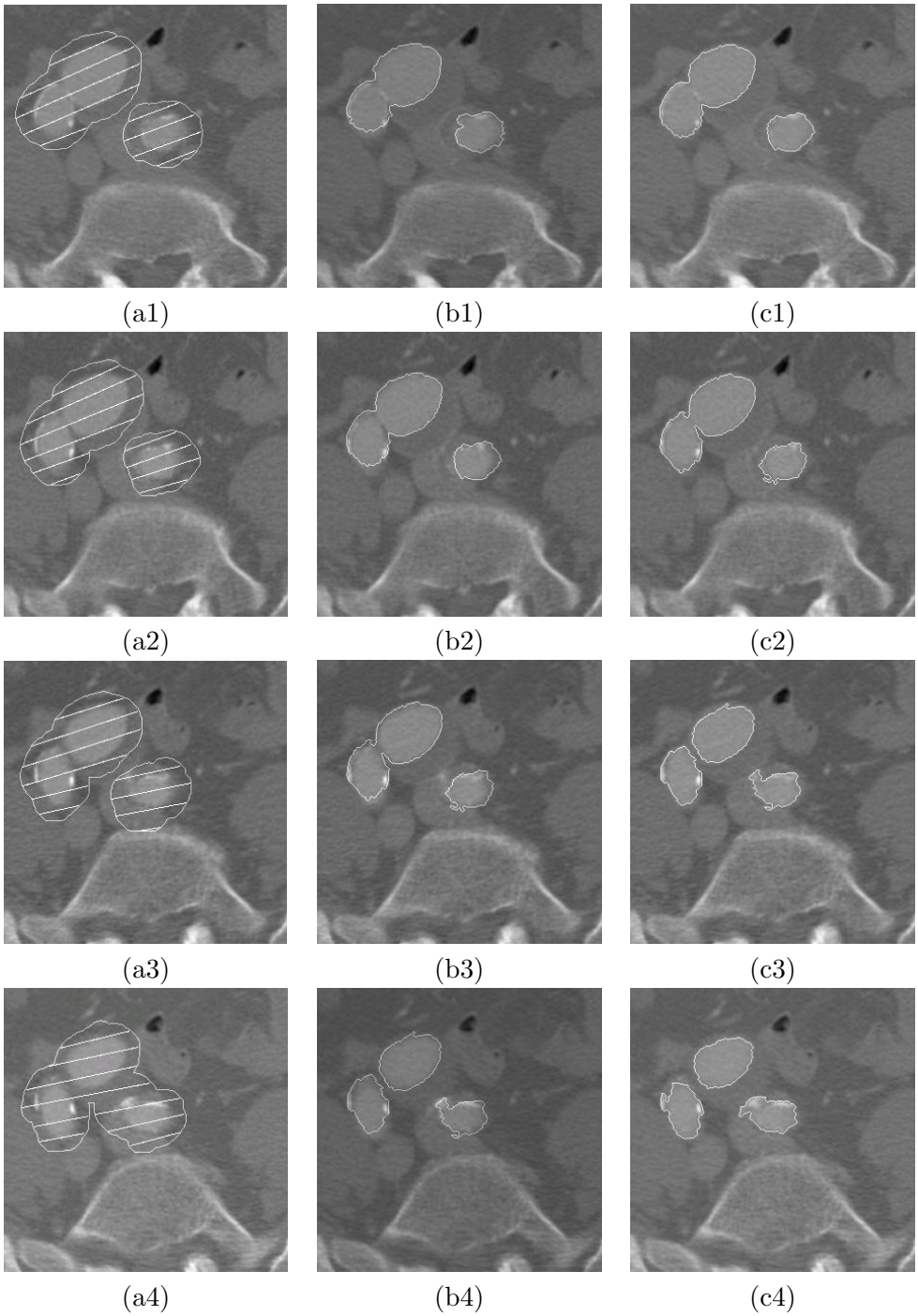
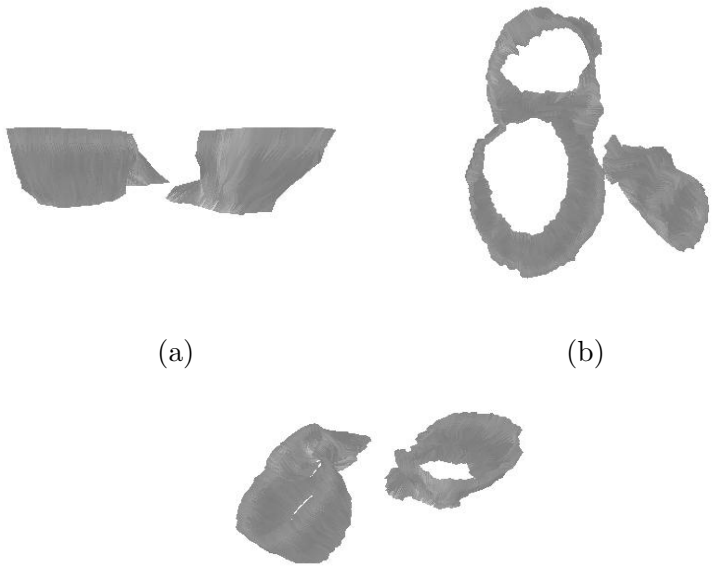


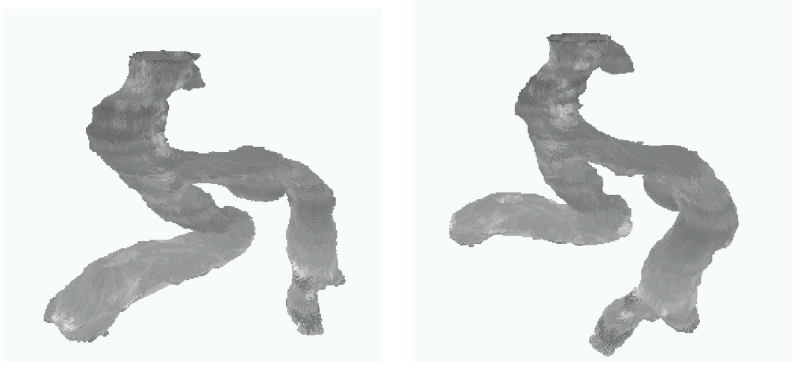
FIG. 31 – *a) RI ω_n, b) Initialisation par les contours précédents sur les quatrième, cinquième, sixième et septième coupe de la séquence. c) Convergence des contours obtenus.*



(a) (b)

(c)

FIG. 32 – *a) Reconstruction 3D d'une partie de l'aorte à partir des contours finaux des sept coupes de la sous-séquence, b) et c) Deux autres vues de la reconstruction 3D.*



(a) (b)

FIG. 33 – *a) et b) Deux vues de la reconstruction 3D de l'aorte obtenue à partir des contours finaux des* 69 *coupes de la séquence.*

L'algorithme complet de suivi est :

Segmentation : de l'image ($n = 1$), $\Rightarrow$ niveau zéro de la courbe $u_{n=1} \Rightarrow$ contour 1
Calculer : dilatation du contour 1 $\Rightarrow$ RI ω_2
Estimation_locale_robuste : du modèle de l'image dans ω_2 pour l'image ($n = 2$)
Segmentation : de l'image ($n = 2$), Init. avec $u_{n=1}$ (contour 1) $\Rightarrow u_{n=2}$(contour 2)
Tant que la séquence n'est pas finie $n <= N$ (N est le nombre d'images)
Localisation : $u_{n/n-1}$, $\sigma^1_{n/n-1}$
RI : dilatation de $\sigma^1_{n/n-1}$ pour obtenir ω_n
Estimation_locale_robuste : du modèle de l'image dans ω_n, pour l'image n
Segmentation de l'image n**:** initialisation avec u_{n-1} (contour $n-1$)

2.2.4 Conclusion et perspectives

Dans cette partie, nous avons proposé une nouvelle méthode de suivi du contour externe d'un organe anatomique à partir d'une séquence d'images médicales. Le contour est initialisé par l'utilisateur uniquement sur la première coupe de la séquence. Notre méthode nous permet d'obtenir une reconstruction 3D d'un organe.

Un nouvel algorithme de suivi dans une séquence d'images médicales a été présenté dans cette partie. Cette méthode est basée sur la segmentation par contours actifs et sur la localisation automatique d'objets. L'algorithme de segmentation a été implémenté avec des méthodes de courbes de niveaux. Il permet le changement de topologie et présente un schéma numérique stable.

La méthode que nous proposons peut avoir des erreurs, mais la propagation de ces erreurs ne perturbe pas l'estimation robuste. En effet, si la région d'intérêt était obtenue par une simple dilatation (β), le précédent contour peut détecter du bruit comme étant une partie entière de l'objet. Ces erreurs peuvent produire des anomalies de détection des objets dans les images suivantes. La prédiction dynamique de déplacement rend possible de localiser un objet et également de déterminer la RI. La prédiction dynamique nous permet de mieux adapter l'estimation de l'objet et limite la propagation des erreurs. Nous avons montré que le terme de régularisation λ pouvait être obtenu par un docteur spécialisé en imagerie médicale afin d'obtenir une courbe lisse.

Pour estimer le modèle de l'image, l'algorithme de localisation est basé sur la prédiction dynamique de déplacement et la détermination de la RI. Les déplacements entre deux coupes successives de la séquence sont obtenus en utilisant également des fonctions de courbes de niveaux. Pour améliorer l'estimation locale robuste du modèle de l'image, nous

avons défini une fonction de courbes de niveaux intermédiaire. L'utilisation de cette fonction de courbes de niveaux intermédiaire améliore la localisation des objets désirés. La méthode de suivi que nous avons présenté est plus automatique que celle utilisée couramment dans les services d'imagerie médicale.

Pour réduire le temps de calcul de la méthode de propagation, il serait possible de paralléliser l'algorithme de détection de contours. Afin d'atteindre cet objectif, nous pouvons utiliser différentes méthodes. Il est possible d'initialiser deux contours sur la première et la dernière image de la séquence. Dans ce cas, la propagation des contours se dirige vers la coupe centrale. Une autre solution serait d'initialiser le contour sur l'image centrale et de le propager dans les deux directions possibles.

Avec les contours obtenus, nous avons montré qu'il était possible d'obtenir une reconstruction 3D d'un organe anatomique. Cette reconstruction constitue une aide importante pour le télé-diagnostic des pathologies de l'aorte abdominale et pour un important nombre de techniques médicales utilisant des images 3D [1].

Notre méthode peut être utilisée pour d'autres organes pour des applications médicales mais également pour du suivi d'objets dans des séquences temporelles [Djemal 04] pour des environnements de robots mobiles par exemple.

1. Nous voulons remercier les Professeurs Agen et Passail, responsables des services de radiologie du C.H.I de Toulon et du CHI de Fréjus Saint Rapahël pour leurs nombreuses discussions, analyses et critiques.

Chapitre 3

Protection des données par insertion de données cachées dans des images

3.1 Introduction

La mise en place d'interfaces de visualisation à distance connaît actuellement une forte demande dans le cas de transfert de données textuelles et images. Le premier problème rencontré concerne la qualité des données transmises. En effet, pour des raisons de temps de transfert au travers du réseau, toutes les données, et en particulier les images, sont comprimées. Le deuxième problème concerne l'aspect sécurité. La sécurisation des images devient extrêmement importante pour de nombreuses applications comme les transmissions confidentielles, la vidéo surveillance et les applications militaires et médicales. Par exemple, la nécessité d'un diagnostic rapide et sûr est vital dans le monde médical [Bernarding 01, Norcen 03]. Pendant le transfert, dans certaines applications, il ne faut absolument pas qu'une image soit dissociée des informations textuelles. De plus, pour des raisons de confidentialité, ces données doivent être rendues illisibles et non déchiffrables, donc cryptées. Dans ce chapitre, je présente des nouvelles méthodes d'insertion de données cachées (IDC) robustes à la compression. Dans le cadre de la protection de données la capacité d'insertion est relativement importante et peut atteindre 10% de la taille de l'image support. Afin d'être robuste à la compression, nous nous sommes orientés vers des approches basées sur la DCT (Discrete Cosinus Transform). Dans le domaine fréquentiel les possibilités d'insertion de données sont nombreuses mais dépendent principalement du lieu d'insertion pouvant varier de la composante continue jusqu'aux très hautes fréquences. Concernant le choix des fréquences, nous avons opté pour une insertion au niveau des basses fréquences ou de la

composante continue. De ce fait, la marque est plus robuste aux diverses transformations que l'image peut subir (compression plus importante, lissage, bruit et rehaussement de contraste). Par contre, le fait d'insérer les données dans les basses fréquences dégrade plus l'image. Pour cela, nos travaux de recherche ont consisté à trouver des méthodes permettant de dégrader le moins possible la qualité de l'image. Dans cette partie nous proposons donc une méthode inductive d'IDC combinant le domaine fréquentiel avec le domaine spatial. Nous montrons que la qualité de l'image est meilleure que dans le cas d'une approche classique d'IDC. Afin de résister à des attaques désynchronisantes (translation, rotation, découpage et changement d'échelle), je propose également, dans ce chapitre, une approche d'insertion de données cachées basée sur le contenu. Cette approche permet d'insérer des données particulières dans chaque région d'intérêt contenue dans l'image.

Dans la section 3.2, je présente les grandes classes d'IDC. Dans la section 3.3, j'approfondis des méthodes d'IDC basées sur la DCT. Section 3.4, je développe une nouvelle méthode d'IDC combinée avec JPEG permettant d'améliorer la qualité des images marquées et comprimées par rapport aux méthodes classiques. Dans la section 3.5, je présente une analyse quantitative théorique et expérimentale de l'amélioration de la qualité des images marquées par la méthode proposée. Dans la section 3.6, j'étends cette méthode aux images couleurs et je propose de m'appuyer sur le contenu des images pour effectuer l'IDC.

Ces travaux ont été développés avec **G. Lo-Varco** dans le cadre de son stage de DEA et de sa thèse ainsi qu'avec **JL. Toutant** dans le cadre de son stage de DEA et de sa thèse et **Ph. Amat** dans le cadre de son stage de DEA. Cette partie a donné lieu aux publications suivantes : [Puech 01c, Puech 01d, Puech 02, Lovarco 03a, Lovarco 03b, Puech 03, Lovarco 04b, Rodrigues 04b, Lovarco 04a, Lovarco 05c, Lovarco 05a, Lovarco 05b, Toutant 05a, Toutant 05b, Amat 05].

3.2 Insertion de données cachées dans des images

Actuellement, la transmission d'images est une nécessité. Afin de sécuriser leur transfert sur les réseaux, deux grandes familles de technologies se développent. La première s'appuie sur une protection via le cryptage des données. Plusieurs méthodes ont été développées pour crypter des images [Chung 98, Chang 01, Sinha 03]. Dans ce premier groupe, le décryptage des données nécessitent d'avoir une clef. Nous reviendrons sur les

méthodes de cryptage des images chapitre 4. La seconde famille base la protection sur l'insertion de données cachées (IDC), qui a pour but d'insérer de manière secrète un message à l'intérieur des données [Shih 03]. Ces deux approches peuvent être complémentaire l'une de l'autre.

Les méthodes d'IDC peuvent être une solution afin de sécuriser la transmission des images. Selon l'application, l'objectif de l'IDC est d'insérer de manière invisible dans l'image une marque ou un message. La longueur du message inséré dans l'image peut être relativement importante. L'insertion peut être effectuée de différentes manières en fonction de la longueur du message de la robustesse désirée. Les méthodes d'IDC peuvent être classées de différentes manières [Delaigle 98, Duric 01, Petitcolas 99]. Classiquement, pour les images, il y a deux familles de méthodes d'IDC. Les méthodes qui utilisent le domaine spatial [Bender 96, Nikolaidis 98] et l'autre groupe celles qui utilisent le domaine fréquentiel et en particulier le domaine DCT [Bors 98, Chang 02, Tseng 04]. Des méthodes hybrides ont également été envisagées [Shih 03].

3.2.1 Stéganographie, tatouage et insertion de données cachées

Le terme stéganographie provient d'une racine latine signifiant *écriture cachée*. Dans l'antiquité, déjà, une volonté d'échanger des informations en dissimulant la communication aux yeux des autres existait. Herodotus (486 - 425 avant JC) rapporte des anecdotes à ce sujet. Ainsi, un message tatoué sur le crâne rasé d'un esclave était transmis secrètement après la repousse de ses cheveux. De même, l'information secrète n'était pas inscrite sur la cire des tablettes de scribe, mais sous celle-ci, sur la planche de bois servant de support. L'encre invisible et des techniques plus évoluées sont venues enrichir le domaine. Des prisonniers de guerre notamment cachaient des messages codés en morse dans leurs missives à l'aide des points et des tirets de lettres telles que les *i*, *j*, *t* et *f*. L'inconvénient de telles méthodes est que le texte servant d'alibi à l'échange était difficile à construire et donc éveiller les soupçons. Ces applications anciennes n'ont néanmoins pas inspirées les chercheurs actuels : la stéganographie n'a fait son apparition dans le numérique que très récemment. Le monde de la recherche et l'industrie s'est massivement intéressé à la cryptographie jusque dans les années 90. Il a fallu attendre 1996 pour que la première conférence académique dans le domaine ait lieu. Les intérêts liés à la propriété intellectuelle ont permis ce regain d'intérêt.

A l'instar de la cryptographie, la stéganographie assure une sécurisation de l'information. La différence se situe au niveau des moyens mis en oeuvre : la cryptographie protège

le contenu du message, la stéganographie le cache. En cela, elle possède un avantage : un message crypté fournit l'information d'un échange secret; au contraire un message caché au sein d'un document numérique ne laisse rien paraître de la communication sous-jacente.

Aujourd'hui, la stéganographie se réfère au camouflage d'une information sensible dans une information banale. Nous allons nous intéresser à la dissimulation au sein d'images, mais tous les médias numériques, vidéos, sons et fichiers textes, peuvent servir de support. Trois points permettent de caractériser une méthode de stéganographie :

- Sa capacité : la quantité d'information dissimulable,
- Sa sécurité : le degré d'invisibilité de l'information cachée,
- Sa robustesse : la résistance de l'information insérée à différents types de traitement de l'image, allant de l'ajout d'un bruit gaussien à des rotations de l'image.

Ces trois points sont dépendants : plus la quantité d'information intégrée est importante, moins l'invisibilité et la robustesse de la méthode sont préservées [Petitcolas 99].

3.2.2 Caractérisation de l'IDC par sa méthode d'extraction

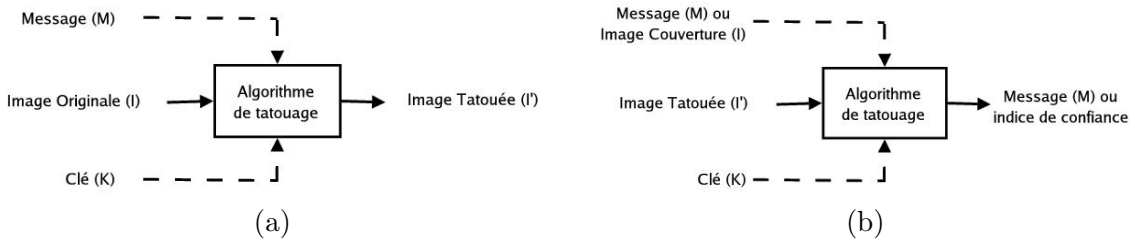


FIG. 34 – *(a) Schéma générique de marquage d'images, (b) Schéma générique d'extraction du message.*

L'insertion et l'extraction du message suivent toujours les schémas présentés figure 34. Les informations nécessaires à l'extraction vont caractériser l'IDC :

- L'IDC est dite non aveugle ou privée quand l'image originale est nécessaire à l'extraction. Ce n'est que grâce à sa comparaison avec l'image marquée que pourra être extrait le message secret.
- L'IDC est dite aveugle dans le cas contraire. Le message peut être extrait par la connaissance de la méthode d'IDC et éventuellement d'une clé.
- L'IDC est dite asymétrique quand l'extraction du message ne nécessite aucun secret. Le marquage est fait sans clé ou selon un schéma analogue à la cryptographie asymétrique, avec clé publique pour le cryptage et clé privée pour l'extraction.

Le choix d'une de ces méthodes dépend de l'utilisation à laquelle est destinée l'application.

3.2.3 Les grandes classes d'IDC

3.2.3.1 L'IDC dans le domaine spatial

L'IDC dans le domaine spatial est le premier à avoir été utilisé et le plus simple à mettre en place. Il se base sur la représentation des images sous forme de matrices de pixels. Les bits du message à dissimuler sont simplement substitués aux bits de poids faibles (Less Significant Bit, LSB) des pixels de l'image [Bender 96, Nikolaidis 98, Chan 04].

L'avantage de cette méthode, outre sa simplicité, est sa grande capacité : elle permet de dissimuler un message d'une taille, en bits, de l'ordre de la résolution en pixels de l'image servant de couverture. Par exemple, une image d'une dimension de 512×512 pixels en niveau de gris (8 bits) peut contenir une dizaine de pages dactylographiées. Les variations sur les intensités des pixels sont très faibles et permettent une invisibilité importante. Les faiblesses de telles méthodes apparaissent au niveau de leur robustesse. Un premier problème vient de la localisation de l'information : un bit du message est associé à un pixel unique de l'image. Le deuxième problème est l'importance négligeable des modifications apportées. L'insertion d'un bit sur le LSB d'un pixel est invisible, mais sa suppression l'est tout autant.

Ces méthodes ont été améliorées en utilisant un générateur de nombres pseudo-aléatoires (GNPA) afin d'utiliser une clef pour avoir accès à l'information insérée. Le GNPA diffuse dans le message toute l'image et rend plus difficile la stéganalyse [Fridrich 02a]. Par exemple un algorithme très simple consiste à insérer un message M composé de k bits b_i avec $M = b_1 b_2 ... b_k$ dans une image de N pixels. Nous devons calculer le facteur d'insertion $E_f = N/k$, puis diviser l'image en zones de taille E_f. Chaque zone est utilisée pour insérer un bit b_i du message. Cette procédure garantit que le message est bien dispersé dans toute l'image. En utilisant un GPNA nous choisissons aléatoirement une zone et un pixel $p(n)$ de cette zone afin de modifier son LSB :

$$p(n) = p(n) - p(n)\ mod(2) + b_i. \tag{35}$$

Il est possible de contourner en partie ces inconvénients en introduisant l'information de manière redondante, par exemple. Mais il semble plus judicieux de se tourner vers d'autres méthodes un peu plus évoluées, comme le SSB-4(Stéganographie par substitution du bit 4) [Rodrigues 04b].

La prise en compte des limites du système visuel humain permet un marquage spatial plus robuste, le SSB-4. L'oeil est incapable de discerner des variations d'intensité inférieures à 4 niveaux de gris. L'insertion ne va donc plus s'effectuer sur le LSB et par substitution, mais sur un bit de poids plus fort et par approximation de la valeur originale. Le bit 4 va ainsi prendre la valeur du bit à insérer et les autres bits du message vont être modifiés pour s'approcher au mieux possible de l'intensité initiale. La dégradation de l'image va rester inférieure au seuil de détection de l'oeil humain, mais offrir un marquage plus robuste. Cette amélioration a un coût pour l'invisibilité du message : l'étude des histogrammes de couleurs montrent des pics caractéristiques.

	Valeur binaire							
Décimal	8	7	6	5	**4**	3	2	1
46	0	0	1	0	**1**	1	1	0
48	0	0	1	***1***	**0**	***0***	***0***	0

TAB. 2 – *Exemple d'IDC selon la méthode SSB-4.*

Le tableau 2 montre un exemple d'application du SSB-4. Ce marquage ne s'applique que pour des intensités de pixels comprises entre 4 et 251. Dans les autres cas, il induirait des modifications supérieures à 4 niveaux de gris.

3.2.3.2 L'IDC dans le domaine fréquentiel

L'IDC dans le domaine spatial localisant trop l'information nous proposons d'étudier des méthodes travaillant dans le domaine fréquentiel. L'image peut être vue comme un signal qui se représente alors comme une somme de fonctions périodiques à deux dimensions. Transformée de Fourier, Transformée Cosinus Discrète, ondelettes, les possibilités pour obtenir une telle représentation de l'image ne manquent pas. Néanmoins, la forme globale du résultat ne change pas. Nous obtenons des basses fréquences qui définissent l'allure général de l'image et des hautes fréquences qui viennent compléter les premières en ajoutant du détail. L'IDC fréquentielle est plus robuste que l'IDC spatiale pour nombre de traitements, notamment la compression si celle-ci se base sur la même transformée. La capacité offerte, tout comme l'invisibilité et la robustesse, dépendent des orientations choisies : un marquage sur les hautes fréquences va peu modifier l'image tandis qu'un marquage sur les basses fréquences sera plus résistant

3.2.3.3 D'autres méthodes d'IDC

Il est un peu restrictif de limiter les méthodes d'IDC aux deux classes, spatiale et fréquentielle. Beaucoup d'autres aspects ont été étudiés [Deguillaume 02, Hartung 99, Voyatzis 99] et ont conduit à des marquages utilisant les caractéristiques des images couleurs [Lovarco 04a] ou des textures. Des techniques s'appliquent même aussi bien aux deux classes présentées, comme l'IDC par étalement de spectres ou celui par modification du code fractal.

L'IDC par étalement de spectre est une technique de dissimulation apparue dans les communications radio pendant la seconde guerre mondiale. Un signal à bande étroite est diffusé sur une largeur de bande plus importante pour rendre l'énergie du signal sur chaque fréquence trop faible pour être détectée. Dans le cas de l'IDC, comme son énergie est répartie, le message reste accessible même si des modifications ont détruit cette énergie en certains endroits. Cette méthode d'IDC est définie plus formellement par une superposition linéaire de fonctions bidimensionnelles, chacune représentant un bit du message. L'image marquée est obtenue en ajoutant ce signal à l'image. Une analyse statistique du signe de la démodulation du signal permet la récupération du message. Il faut bien sûr connaître les fonctions bidimensionnelles utilisées pour l'insertion. La capacité offerte est limitée puisque la robustesse du message est assurée par une certaine redondance de l'information [Cox 97a, Winkler 99].

L'IDC par modification du code fractal se base sur un schéma de compression fractale qui transforme l'image originale en un code fractal de taille inférieure. Il se construit en cherchant des systèmes de fonctions itérées ou IFS. L'image est décomposée en blocs 8×8. Le code fractal consiste alors en un appariement des blocs selon leurs similarités, c'est-à-dire la minimisation de leur erreur quadratique moyenne. Il est rare de trouver des blocs similaires (erreur quadratique moyenne nulle) dans une image ordinaire et la signature va être insérée en ajoutant de telles similarités artificiellement et de manière invisible [Bas 98].

3.2.4 Évaluation de l'IDC

3.2.4.1 Qualité d'une IDC

La qualité d'une IDC est en général associée à la ressemblance entre l'image marquée et l'image originale. Plus le support est modifié, moins bonne est la qualité. L'importance des perturbations étant principalement liée à la quantité d'information dissimulée, la qualité de l'insertion va déterminer sa capacité. Des indicateurs sont nécessaires pour évaluer cette

qualité. Ils se divisent en deux types, les indicateurs subjectifs et les indicateurs objectifs.

Les premiers se basent sur les limites du système visuel humain (SVH). L'invisibilité à l'oeil est une première approximation de la qualité d'une IDC. Voici quelques résultats qui sont à prendre en compte :

- l'oeil ne peut pas détecter une variation d'intensité entre deux pixels inférieure à 4 (cas du codage de l'intensité sur 8 bits),
- l'oeil est plus sensible aux variations dans les zones homogènes que dans les zones texturées.

En complément de ces méthodes subjectives, des mesures objectives sont aussi utilisées : histogrammes des intensités de pixels, valeurs moyennes et moments. Ces méthodes ne s'intéressent pas aux aspects visuels de l'image, mais plutôt à une étude statistique de sa représentation. Deux indicateurs sont particulièrement utilisés, l'erreur quadratique moyenne (EQM) et le pic du rapport signal à bruit (Peek Signal Noise Ratio, PSNR).

L'EQM permet de quantifier les différences entre deux images. C'est un calcul de distance entre deux matrices. De manière générale :

$$EQM = \frac{\sum_{\imath=0}^{N-1}(p(\imath) - p'(\imath))^2}{N}, \tag{36}$$

avec N, le nombre de pixels des images, $p(i)$ et $p'(i)$ respectivement les intensités des pixels d'index i de chacune des images.

Le PSNR donne une idée plus précise de la dégradation d'une l'image. Il s'évalue selon la formule suivante :

$$PSNR = 10log_{10}\frac{255^2}{EQM}. \tag{37}$$

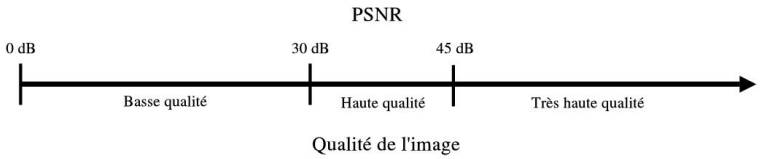


FIG. 35 – *Correspondance entre PSNR et qualité objective d'image.*

Deux images identiques vont donner un PSNR infini. De manière pratique, nous pouvons nous reporter à l'échelle quantitative de la figure 35.

3.2.4.2 La stéganalyse

La stéganalyse s'attache justement à évaluer la sécurité des méthodes d'IDC. Elle analyse les caractéristiques inhabituelles ou les dégradations apportées au médium par l'IDC et

attaque le message caché. Différentes attaques peuvent être produites visant la détection, l'altération, voir même l'extraction de l'information cachée [Johnson 98, Silman 01].

La détection n'est pas facile. Certaines méthodes laissent des artefacts qu'une analyse statistique de l'image comme son histogramme de couleurs peut mettre en évidence. Mais l'identification de tels motifs caractéristiques n'est souvent possible que par les comparaisons de plusieurs jeux d'images originales et marquées associées. Ces motifs peuvent se caractériser par les palettes de couleurs utilisées, les relations entre les couleurs ou encore un bruit exagéré.

Il est plus facile d'altérer ou même de détruire le message. Dans le cas courant d'un tatouage spatial sur les bits de poids faibles, une légère compression de l'image suffit. Dans la littérature, la compression, l'ajout de bruit gaussien, l'utilisation de certains filtres ont souvent été mis en avant pour montrer la robustesse des méthodes. Néanmoins des distorsions spécifiques, comme la rotation, même de très faible importance, suffisent à rendre le message illisible.

3.2.5 Applications industrielles de l'IDC

Malgré son avènement tardif, l'IDC touche aujourd'hui un grand nombre de domaines. Tous n'ont pas les mêmes besoins en terme de capacité, d'invisibilité et de robustesse. L'industrie musicale et cinématographique, à l'origine du regain d'intérêt pour la stéganographie, se focalise plutôt sur la robustesse. Le but poursuivi est l'insertion d'un message indélébile dans les documents numériques pour assurer la propriété. Un tel marquage pourrait servir pour les DVD et ainsi les protéger contre la copie. La nouveauté proviendrait de la présence de la protection, non pas sur le support physique, mais intégrée dans le contenu. A l'opposé, la fragilité d'une marque va être utile pour l'authentification de documents. La moindre modification subit par le médium va la détruire et faire perdre sa valeur au document. Cet aspect peut facilement trouver des applications dans des échanges contractuels voir même pour des élections numériques. Bien sûr, la facette invisibilité de la stéganographie va intéresser tout le domaine du renseignement, des institutions militaires au terrorisme, en passant par le monde de l'industrie. La capacité d'insertion est importante pour ces applications, ce n'est plus seulement une marque qui est cachée, mais un message beaucoup plus conséquent qui est transmis. Il va de soi que l'aspect stéganalyse est aussi très prisé dans ce domaine.

L'IDC est un domaine jeune et vaste. Même si elle présente des limitations semblables à celle de la cryptographie, elle a encore une évolution importante à venir. Entre une

application destinée à l'authentification de document qui présente une marque fragile et une autre de dissimulation d'information de taille importante les objectifs sont différents. Il paraît alors difficile de définir une méthode universelle. Il semble plus judicieux et plus aisé de les spécialiser pour des types d'applications bien précis. Dans notre cas, nous cherchons à développer une IDC résistant à la compression JPEG. Nous proposons, section suivante, de nous intéresser à ce standard de compression et aux méthodes existantes.

3.3 IDC basées sur la DCT

Dans la section 3.3.1 je présente les différentes étapes de l'algorithme JPEG pour ensuite détailler, section 3.3.2, des méthodes d'IDC basée sur la DCT.

Le format d'image JPEG est un format de compression fortement utilisé pour les images numériques. Bon nombre de méthodes d'IDC s'appuient sur ses caractéristiques pour lui être robuste. Elles utilisent donc la transformée cosinus discrète (Discrete Cosine Transform, DCT) pour obtenir une représentation fréquentielle de l'image.

3.3.1 L'algorithme de compression JPEG

La figure 36 présente le fonctionnement de la compression JPEG. Elle s'appuie sur la DCT pour passer dans le domaine fréquentiel. Ensuite, l'image est dégradée par l'étape de quantification avant d'être compressée [Wallace 91, Pennebaker 93].

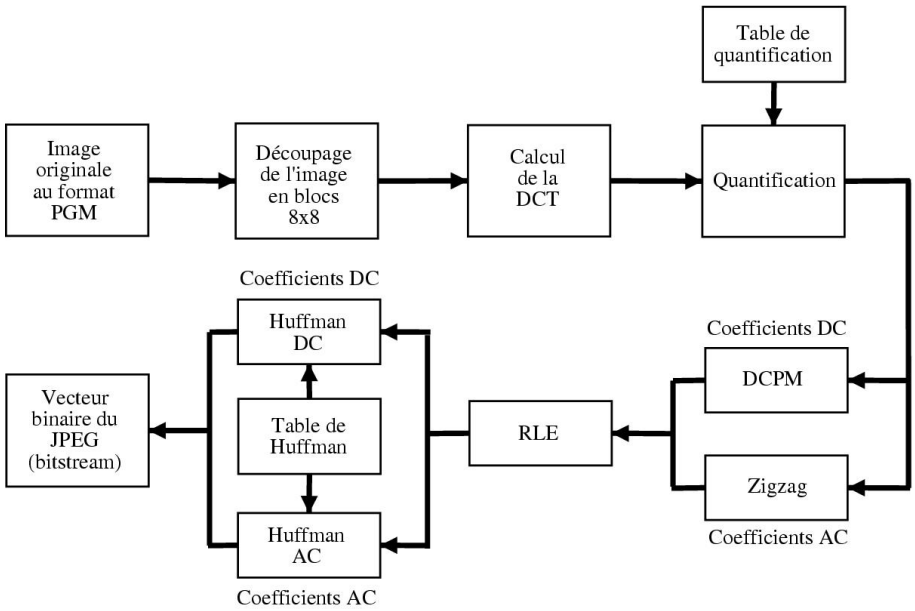


FIG. 36 – *l'algorithme de compression JPEG.*

3.3.1.1 La Transformée Cosinus Discrète

La DCT permet d'obtenir une représentation fréquentielle particulière de l'image à partir de sa matrice de pixels. A partir des intensités des pixels $p(i,j)$, nous obtenons les coefficients DCT associés $F(u,v)$, équation (38). Il existe bien la transformée inverse, la Transformée Cosinus Discrète Inverse (Inverse Discrete Cosine Transform, IDCT) permettant de revenir au domaine spatial à partir de la connaissance des coefficients DCT, équation (39) :

$$F(u,v) = \frac{2}{n}C(u)C(v)\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}p(i,j)cos\left(\frac{\pi(2i+1)u}{2n}\right)cos\left(\frac{\pi(2j+1)v}{2n}\right) \quad (38)$$

$$p(i,j) = \frac{2}{n}\sum_{u=0}^{n-1}\sum_{v=0}^{n-1}C(u)C(v)F(u,v)cos\left(\frac{\pi(2i+1)u}{2n}\right)cos\left(\frac{\pi(2j+1)v}{2N}\right), \quad (39)$$

avec :

$$\begin{cases} C(x) = \frac{1}{\sqrt{2}} \text{ si } x = 0 \\ \\ C(x) = 1 \text{ si } x \neq 0 \end{cases},$$

avec le support supposé de taille $n \times n$.

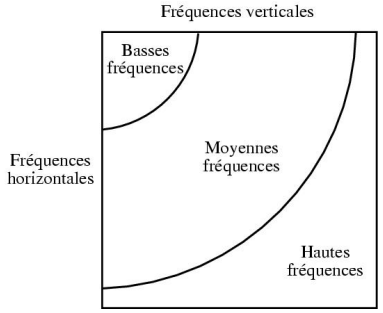


FIG. 37 – *Répartition des fréquences dans la matrice des coefficients DCT.*

Nous pouvons faire différentes remarques sur les termes qui interviennent :

- les deux cosinus définissent la fréquence du signal respectivement dans chacune des deux directions du plan de l'image. Ainsi, les basses fréquences vont se situer dans la partie supérieure gauche de la matrice des coefficients DCT et les hautes fréquences dans la partie inférieure droite comme présenté figure 37.
- Les coefficients DCT, $F(u,v)$, définissent l'amplitude des signaux. Pour limiter leur grandeur, les intensités des pixels définies entre 0 et 255, sont ramenées autour de 0, entre -128 et 127.

– le coefficient DCT, $F(0,0)$ est appelé composante continue ou composante DC (Direct Current). Il possède la propriété d'être proportionnel à la moyenne des intensités des pixels sur le bloc considéré :

$$\begin{aligned} F(0,0) &= \frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} p(i,j).cos\,(0)\,.cos\,(0) \\ &= nE\left[p(i,j)\right]. \end{aligned}$$

Son utilisation sur des images de grandes dimensions va s'avérer coûteuse. Elle est donc généralement appliquée sur des fractions de l'image, des blocs de 8×8 pixels. Cette dimension permet un bon compromis entre le maintien d'une bonne corrélation entre les pixels voisins et la rapidité du calcul. L'augmentation de la taille des blocs permet une meilleure compression, au détriment du temps de traitement.

3.3.1.2 La quantification : étape de la perte de l'information

La quantification représente la phase non conservatrice du processus de compression JPEG. Elle divise chaque coefficient DCT, $F(u,v)$ par un nombre $Q(u,v)$, appelé quantum ou coefficient de quantification, fixé par une matrice de même dimension que les blocs utilisés, la matrice de quantification :

$$F'(u,v) = \left[\frac{F(u,v)}{Q(u,v)}\right], \tag{40}$$

avec $[x]$, la valeur entière la plus proche de x.

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

FIG. 38 – *Matrice de quantification standard pour un facteur de qualité de 50%.*

Chaque matrice de quantification est construite à partir de la matrice de quantification standard, présentée figure 38, et d'un facteur de qualité, Q. Ce facteur compris entre 1% et 100% définit la tendance entre qualité de l'image et compression. Le taux de compression n'est en effet pas accessible à l'avance, il est dépendant du contenu de l'image. Même

pour un facteur de qualité de 100% (les coefficients de quantification sont alors tous à 1), le JPEG provoque une perte d'information puisque les coefficients DCT quantifiés sont conservés sous la forme d'entiers. La valeur des quanta est d'autant plus élevée que les coefficients DCT correspondants participent moins à la qualité de l'image. Les matrices de quantification présentent donc généralement une progression le long de leur diagonales descendantes : la quantification croît à mesure que les fréquences augmentent. Cet accroissement définit le facteur de qualité. Dans le cas d'une image couleur, pour les deux plans de chrominance, les coefficients de la matrice de quantification sont plus grands que ceux utilisés pour le plan de luminance.

3.3.1.3 Codage entropique

La quantification dégrade les données, l'information restante va maintenant être compressée. Deux éléments sont distingués : les composantes DC et les autres coefficients DCT quantifiés appelés composantes AC (Alternating Current).

Les composantes DC de blocs adjacents sont fortement corrélées puisque elles représentent l'intensité moyenne des pixels de leurs blocs respectifs. Au lieu de conserver leurs valeurs, généralement importantes, ce sont leurs différences relatives qui sont conservées.

Cette modification effectuée, les blocs de l'image sont parcourus de gauche à droite et de bas en haut. Dans chacun d'eux, l'ordre des coefficients sera défini par un parcours en zigzag qui ordonne les fréquences de manière croissante. Le résultat est un vecteur de tous les coefficients de tous les blocs de l'image. Le codage Huffman s'applique sur ce vecteur pour réduire sa taille en codant les événements sous formes binaires. Les séries de 0, présentes au niveau des hautes fréquences grâce à la quantification, vont être particulièrement intéressantes pour le taux de compression.

3.3.2 Les méthodes d'IDC robustes à la compression JPEG

3.3.2.1 IDC sur la composante continue

Le coefficient DCT $F(0,0)$ est proportionnel à la valeur moyenne de l'intensité des pixels du bloc de l'image auquel il se rapporte. Il présente un intérêt particulier pour l'IDC puisque les perturbations qu'il va induire sont parfaitement définies : il provoque une modification homogène de l'intensité des pixels [Upham 97]. Il est aussi le coefficient associé à la plus basse fréquence, donc le moins susceptible d'être dégradé par des modifications de l'image, un point intéressant vis-à-vis de la robustesse. Enfin, du point de vue de la compression, la composante DC sera rarement nulle, donc son utilisation pour l'insertion

ne pénalise pas le taux de compression.

En résumé, l'IDC sur la composante continue par substitution du bit du message au LSB présente une invisibilité satisfaisante, une bonne robustesse et conserve une compression importante.

3.3.2.2 JPEG-JSTEG

Cette méthode propose une meilleure capacité en se basant sur une remarque simple : en insérant le message sur les coefficients DCT quantifiés non nuls, la dégradation de la compression est négligeable. L'algorithme JPEG-JSTEG propose donc la substitution du bit de poids faible, de chaque coefficient quantifié DCT de valeur absolue strictement supérieure à 1, par un bit du message [HW99]. La capacité de stockage offerte n'est pas fixe, elle dépend de l'image et du facteur de qualité.

3.3.2.3 IDC par modification de la matrice de quantification

Les coefficients les plus importants pour la qualité de l'image sont ceux correspondant aux basses fréquences. Les coefficients les plus utiles pour la compression sont ceux correspondant aux hautes fréquences. L'idée de [Chang 02] est donc de cacher l'information dans les coefficients des moyennes fréquences dont l'influence est moindre sur les deux phénomènes cités.

16	11	10	16	1	1	1	1
12	12	14	1	1	1	1	55
14	13	1	1	1	1	69	56
14	1	1	1	1	87	80	62
1	1	1	1	68	109	103	77
1	1	1	64	81	104	113	92
1	1	78	87	103	121	120	101
1	92	95	98	112	100	103	99

FIG. 39 – *Matrice de quantification modifiée pour un facteur de qualité de 50%.*

Pour conserver une bonne qualité d'image, les coefficients à marquer ne seront pas quantifiés. La matrice de quantification aura l'allure de celle présentée figure 39. Cette méthode va permettre de dissimuler des messages de taille conséquente et surtout la capacité d'insertion est connue à l'avance : 2 bits sont insérés par substitution sur les deux bits de poids faible des 26 coefficients moyens, soit une capacité de 52 bits par bloc.

L'invisibilité de l'image n'est pas optimale. La matrice de quantification est intégrée

dans l'image JPEG et est facilement accessible. Les modifications caractéristiques qui lui sont apportées rendent l'IDC complètement transparente. Le secret ne repose plus que sur le cryptage préalable du message. Il reste illisible, mais peut être facilement altéré. L'autre inconvénient de cette méthode concerne la qualité de l'image et la compression. La substitution de certains coefficients de quantification par 1, va conduire à un facteur de qualité plus élevé que celui annoncé (celui de la matrice de quantification avant transformation). Évidemment la compression suit la même évolution.

3.3.2.4 IDC adaptative basée sur le SVH

Les méthodes d'IDC précédentes permettent l'insertion d'une quantité limitée d'information, surtout quand le facteur de qualité diminue. Ce n'est pas vrai pour la méthode qui modifie la matrice de quantification, mais dans son cas, les dégradations sur l'image se révèlent importantes.

L'IDC adaptative va séparer les blocs 8×8 en fonction de leur homogénéité. Les modifications induites par l'insertion seront alors intégrées principalement dans les blocs hétérogènes, où le système visuel humain (SVH) aura plus de mal à les discerner. Un paramètre de capacité va permettre de définir la relation entre la quantité d'information à cacher et la qualité relative de l'image marquée [Tseng 04].

Le choix des coefficients DCT quantifiés à marquer est le même que pour la méthode JPEG-JSTEG : une fois encore, les coefficients à 0 sont préservés pour garder un bon niveau de compression. Par contre, l'insertion ne se limite plus à un bit par pixel. Pour chaque coefficient DCT quantifié, le nombre de bits utilisables est déterminé en fonction de sa capacité propre et de sa valeur. L'énergie de l'image se repartit principalement dans les basses fréquences. La matrice de quantification standard donne donc une idée de sa répartition. En se basant sur celle-ci et sur le paramètre de capacité, la capacité propre de chaque coefficient peut être évaluée.

Les composantes continues des blocs seront traitées à part. En effet, elles ne produisent pas de dégradations significatives quand le facteur de qualité est élevé, mais conduisent à des déformations importantes quand le facteur de compression est élevé. Cette méthode d'IDC n'apporte pas réellement de qualité objective supplémentaire : les résultats à ce niveau sont en accord avec les autres méthodes. Par contre, la qualité subjective devrait être améliorée.

3.3.2.5 D'autres méthodes d'IDC robustes à JPEG

D'autres méthodes d'IDC ont été développés afin d'être robustes à la compression JPEG [Piva 97, Bors 98, Bors 99]. La plupart de ces méthodes permet surtout d'insérer juste une marque à l'intérieur de l'image de taille maximale de 64 bits. Ces méthodes font partie des méthodes d'IDC appelées tatouage d'images. Un autre schéma, appelé QIM (quantization index modulation), a été également proposé afin d'approcher des valeurs stables et donc plus robustes aux traitements que peuvent subir les images [Chen 01].

3.4 Une nouvelle méthode : l'IDC par bloc avant quantification

3.4.1 Présentation de la méthode

Dans cette section nous décrivons une nouvelle méthode d'IDC basée sur la DCT. Afin d'être robuste au bruit nous proposons d'insérer notre message au niveau de la composante DC de la DCT [Upham 97, Bors 99].

La méthode d'IDC proposée est basée sur la substitution du LSB (Least Significant Bit) de la composante DC, élément (0,0) de la DCT.

Donc, l'objectif est d'insérer un message M composé de k bits $b_\imath$ avec $M = b_1 b_2 ... b_k$. Dans un bloc carré composé de n^2 pixels $p_\imath$ d'une image de N pixels, nous calculons la composante continue $F(0,0)$ de la DCT de ce bloc :

$$F(0,0) = \frac{1}{n} \sum_{\imath=0}^{n^2-1} p_\imath. \tag{41}$$

où n est le coté du bloc de pixels. Pour chaque bloc de l'image nous numérotons les pixels de 0 à $n^2 - 1$.

Nous avons vu section 3.3.1, que classiquement après quantification nous avons :

$$F'(0,0) = \left[\frac{F(0,0)}{Q(0,0)} \right], \tag{42}$$

où $[x]$ est l'entier le plus proche de x, et $Q(0,0)$ est le coefficient de quantification.

Pour insérer notre message, après la quantification nous changeons $F(0,0)$ par $F_{dh}(0,0)$ [1] tel que :

$$F_{dh}(0,0) = \begin{cases} \lfloor \frac{F(0,0)}{Q(0,0)} \rfloor \times Q(0,0) & \text{si } \lfloor \frac{F(0,0)}{Q(0,0)} \rfloor \ mod(2) = b_\imath, \\ \lceil \frac{F(0,0)}{Q(0,0)} \rceil \times Q(0,0) & \text{si } \lceil \frac{F(0,0)}{Q(0,0)} \rceil \ mod(2) = b_\imath, \end{cases} \tag{43}$$

1. *dh* pour Data Hiding

où $\lfloor x \rfloor$ est l'entier juste au dessous de x et $\lceil x \rceil$ est l'entier juste au dessus de x.

A partir de l'équation (43), après quantification nous avons :

$$F'_{dh}(0,0) = \frac{F_{dh}(0,0)}{Q(0,0)}. \tag{44}$$

Nous pouvons remarquer que dans l'équation (44), il n'était plus nécessaire d'arrondir la valeur à l'entier le plus proche puisque le résultat final est déjà un entier.

Au lieu de modifier directement $F(0,0)$ pour obtenir $F_{dh}(0,0)$, dans notre méthode, nous proposons de modifier certains pixels du bloc afin d'obtenir la valeur souhaitée. C'est cette étape qui justifie le terme d'insertion de données cachées par induction puisque même si l'insertion et l'extraction du message sont effectuées dans le domaine fréquentiel la modification, quand à elle, a bien lieu au niveau des pixels. Nous devons donc pour cela analyser le nombre nécessaire de pixels à modifier n_{dh} de ce bloc dans le but d'obtenir la valeur appropriée pour $F_{dh}(0,0)$. Pour cela, nous calculons la différence d entre $F(0,0)$ et $F_{dh}(0,0)$:

$$d = F(0,0) - F_{dh}(0,0). \tag{45}$$

Donc à partir de l'équation (45) nous obtenons n_{dh} le nombre de pixels à modifier de ce bloc :

$$n_{dh} = [|d| \times n]. \tag{46}$$

Nous devons donc changer la valeur de n_{dh} pixels du bloc de n^2 pixels et nous obtenons des nouvelles valeurs de pixels $p_{dh}(\imath)$:

$$p_{dh}(\imath) = p(\imath) - sign(d), \tag{47}$$

où $sign(x) = -1$ si $x < 0$ et $sign(x) = 1$ si $x >= 0$.

Les pixels modifiés sont choisis dans le bloc en fonction d'un critère qui peut être la variance du bloc ou la position spatiale des pixels dans le bloc par exemple. Dans le cas où il y a plus de pixels à modifier que de pixels disponibles dans le bloc, $n_{dh} > n$, (cela dépend de la valeur du coefficient de quantification $Q(0,0)$), nous appliquons l'équation (47) à tous les pixels du bloc et nous répétons l'opération pour les $n_{dh}\%n$ pixels restants.

Pour la composante DC après quantification, nous avons pour chaque bloc l'insertion d'un bit du message :

$$F'_{dh}(0,0) = \frac{1}{n \times Q(0,0)} \Big(\sum_{\imath=0}^{n_{dh}-1} p_{dh}(\imath) + \sum_{\imath=n_{dh}}^{n^2-1} p(\imath) \Big). \tag{48}$$

Pour l'extraction du message, au moment de la décompression de l'image, il suffit alors de lire tous les LSB composantes DC des blocs DCT et de reconstruire le message inséré.

3.4.2 Exemple d'application

Après avoir détaillé les différentes étapes de l'insertion et de l'extraction des données sur la composante DC des coefficients DCT, nous allons les appliquer sur un exemple. Prenons un bloc de 64 pixels d'une image. Sur le plan de luminance, ce bloc est représenté par la matrice 8×8 suivante :

$$M = \begin{bmatrix} 190 & 152 & 136 & 158 & 176 & 191 & 202 & 223 \\ 225 & 212 & 204 & 212 & 220 & 221 & 227 & 235 \\ 244 & 245 & 245 & 245 & 245 & 244 & 243 & 241 \\ 244 & 242 & 243 & 237 & 237 & 242 & 241 & 238 \\ 240 & 230 & 212 & 188 & 193 & 215 & 236 & 225 \\ 223 & 204 & 172 & 133 & 127 & 157 & 196 & 212 \\ 222 & 195 & 152 & 100 & 90 & 125 & 179 & 208 \\ 230 & 206 & 163 & 114 & 109 & 144 & 194 & 215 \end{bmatrix},$$

où chaque coefficient est un niveau de gris d'un pixel.

Nous pouvons alors calculer la composante DC de la DCT suivant l'équation (41) et nous obtenons $F(0,0) = 1608.625$. Le coefficient $F(0,0)$ est alors quantifié par $Q(0,0) = 8$ qui est le premier coefficient de la table de quantification de luminance. Cette valeur de $Q(0,0) = 8$ correspond à une quantification réalisée pour une compression avec un facteur de qualité égal à 75 %. Nous obtenons donc $F'(0,0)/Q(0,0) = 201.078$. Si le bit à insérer est un **1** alors nous devons retenir l'entier inférieur afin d'avoir $\lfloor F'(0,0)/Q(0,0) \rfloor \% 2 = 201\%2 = 1$. Si le bit à insérer est un **0** alors nous devons retenir l'entier supérieur afin d'avoir $\lceil F'(0,0)/Q(0,0) \rceil \% 2 = 202\%2 = 0$. Pour notre exemple choisissons un bit à insérer égal à **1**, dans ce cas nous avons $F_{dh}(0,0) = 201 \times 8 = 1608$ et à partir de l'équation 44 nous obtenons $F'_{dh}(0,0) = 201$. En utilisant l'équation (45), nous pouvons calculer la différence $d = F(0,0) - F_{dh}(0,0) = 1608.625 - 1608 = 0.625$. Nous obtenons alors le nombre de pixels du blocs à modifier $n_{dh} = [|d| \times n] = [0.625 \times 8] = 5$. A partir de l'équation (47), nous en déduisons qu'il faut retrancher 1 à 5 pixels de ce bloc : $p_{dh}(\imath) = p(\imath) - 1$. Ces 5 pixels sont choisis de manière à réduire la variance du bloc. Ainsi, les transformations liées à l'insertion sont moins facilement décelables que dans le cas d'un choix aléatoire des pixels modifiés. Dans notre cas ce sont les 5 pixels d'intensité les plus fortes qui vont être diminués de 1 niveau de gris. Ainsi la variance du bloc est diminuée. La composante continue quantifiée de ce bloc décroît jusqu'à la valeur 201 :

$$\begin{aligned} F'_{dh}(0,0) &= \frac{1}{n \times Q(0,0)} \Big(\sum_{\imath=0}^{n_{dh}-1} p_{dh}(\imath) + \sum_{\imath=n_{dh}}^{n^2-1} p(\imath) \Big) \\ &= \frac{1}{8 \times 8} \Big(\sum_{\imath=0}^{5-1} p_{dh}(\imath) + \sum_{\imath=5}^{63} p(\imath) \Big) \\ &= \frac{F(0,0)}{8} - \frac{5}{64} = 201. \end{aligned}$$

Finalement la matrice marquée M_{dh} est :

$$M_{dh} = \begin{bmatrix} 190 & 152 & 136 & 158 & 176 & 191 & 202 & 223 \\ 225 & 212 & 204 & 212 & 220 & 221 & 227 & 235 \\ \mathbf{243} & \mathbf{244} & \mathbf{244} & \mathbf{244} & \mathbf{244} & 244 & 243 & 241 \\ 244 & 242 & 243 & 237 & 237 & 242 & 241 & 238 \\ 240 & 230 & 212 & 188 & 193 & 215 & 236 & 225 \\ 223 & 204 & 172 & 133 & 127 & 157 & 196 & 212 \\ 222 & 195 & 152 & 100 & 90 & 125 & 179 & 208 \\ 230 & 206 & 163 & 114 & 109 & 144 & 194 & 215 \end{bmatrix},$$

où les valeurs en gras sont les valeurs diminuées de 1 niveau de gris.

Pour extraire l'information insérée, il faut au moment de la décompression du bloc avant l'étape de quantification inverse lire le LSB de la composante DC quantifiée $F_{dh}(0,0)\%2$ afin d'obtenir le bit dissimulé. Dans notre cas, si il n'y a pas eu d'erreur durant la transmission, nous aurons $F_{dh}(0,0)\%2 = 201\%2 = 1$. L'extraction nous fournit donc le résultat attendu. Avec un facteur de qualité égal à 75%, nous avons $Q(0,0) = 8$, par conséquent pour qu'il ne soit plus possible d'extraire l'information il faut que plus de la moitié des pixels ait des niveaux de gris qui varient de 1 dans le même sens. Avec cet exemple nous avons illustré l'aspect inductif de notre méthode d'IDC.

3.4.3 Résultats de la méthode proposée

Dans cette section, nous appliquons notre méthode à l'image Baboon. Nous avons choisi 4 facteurs de qualité (FQ) pour la compression JPEG qui sont 100%, 80%, 70% et 50%, illustrés respectivement figures 40.a1, a2, a3 et a4. Les figures 40.b représentent les images différences entre l'image originale et les images comprimées[2]. Avec la méthode d'IDC proposée nous obtenons les images figures 40.c qui sont fonctions du facteur de qualité pour la compression JPEG. Comme l'image originale est de taille 512×512 pixels, nous pouvons insérer dans l'image 4096 bits, soit 512 caractères ASCII. Les images différences entre l'image originale et les images marquées sont illustrées figures 40.d. Nous constatons, tableau 3, que la dégradation des images est principalement due à la compression. Plus le facteur de qualité diminue, plus l'image est dégradée par la compression et plus l'insertion est également importante. Nous avons également représenté figures 40.e la différence entre l'image comprimée et l'image marquée (également comprimée). Nous constatons que certains blocs ne subissent aucune modification, car la composante continue divisée par le coefficient de quantification est déjà une valeur entière. Au niveau de ces différences, nous

2. Le niveau de gris (ndg) à 128 correspond à une différence nulle, ndg:64=-1, ndg:192=+1, ndg:0=-2, ndg:255=+2.

pouvons également remarquer, tableau 3, que l'EQM de la différence finale plus l'EQM de l'image comprimée est égale à l'EQM de l'image marquée.

	FQ	100 %	90 %	80 %	70 %	50 %	30 %	10 %
Compression	EQM	0.085	13.46	40.71	68.52	116.00	176.16	350.67
	PSNR (dB)	58.82	36.84	32.03	29.77	27.49	25.7	22.68
IDC	EQM	0.100	13.53	40.99	69.29	117.98	181.71	400.79
	PSNR (dB)	58.12	36.82	32.01	29.73	27.41	25.54	22.10
Différence	EQM	0.041	0.18	0.38	0.86	1.99	5.65	50.26
	PSNR (dB)	61.97	55.48	52.36	48.80	45.13	40.61	31.12

TAB. 3 – *EQM et PSNR pour les images comprimées, les images marquées et pour la différence entre les images comprimées et les images marquées pour différents facteurs de qualité de compression.*

Au niveau de la compression, notre méthode d'IDC ne modifie absolument pas le taux de compression par rapport à l'image simplement comprimée par JPEG. Le tableau 4 illustre les tailles des images marquées, l'image originale a pour taille 256.0 ko.

FQ	100 %	90 %	80 %	70 %	50 %	30 %	10 %
Taille (ko)	251.9	119.6	83.9	66.3	48.5	34.8	16.2

TAB. 4 – *Taille des images marquées pour différents facteurs de qualité de compression.*

Quelque soit le facteur de qualité choisi, nous sommes donc capables d'extraire le message inséré. Par contre, plus le facteur de qualité est faible plus l'insertion sera importante. Cependant, si pendant le transfert une compression plus importante est appliquée à l'image alors le message ne résistera pas à cette nouvelle compression. Pour résister à ce type d'attaque, il suffit d'insérer la marque le plus fortement possible (en considérant un FQ=10 % par exemple) même si l'on ne compresse l'image qu'avec un FQ de 100 %. Dans ce cas, l'image au moment de l'IDC est plus fortement dégradée mais complètement robuste à la compression JPEG. Dans la section suivante nous allons généraliser cette méthode d'IDC avant quantification aux méthodes présentées section 3.3.

3.5 Analyse de l'amélioration de la qualité

Dans cette section nous montrons que notre proposition d'IDC avant quantification (section 3.4) peut se généraliser sur les méthodes classiques d'IDC basées sur la DCT dans le but d'améliorer la qualité de l'image marquée.

Toutes les techniques classiques d'IDC basées sur le cycle de production du JPEG présentées dans la section 3.3 font usage d'une insertion après l'étape de quantification

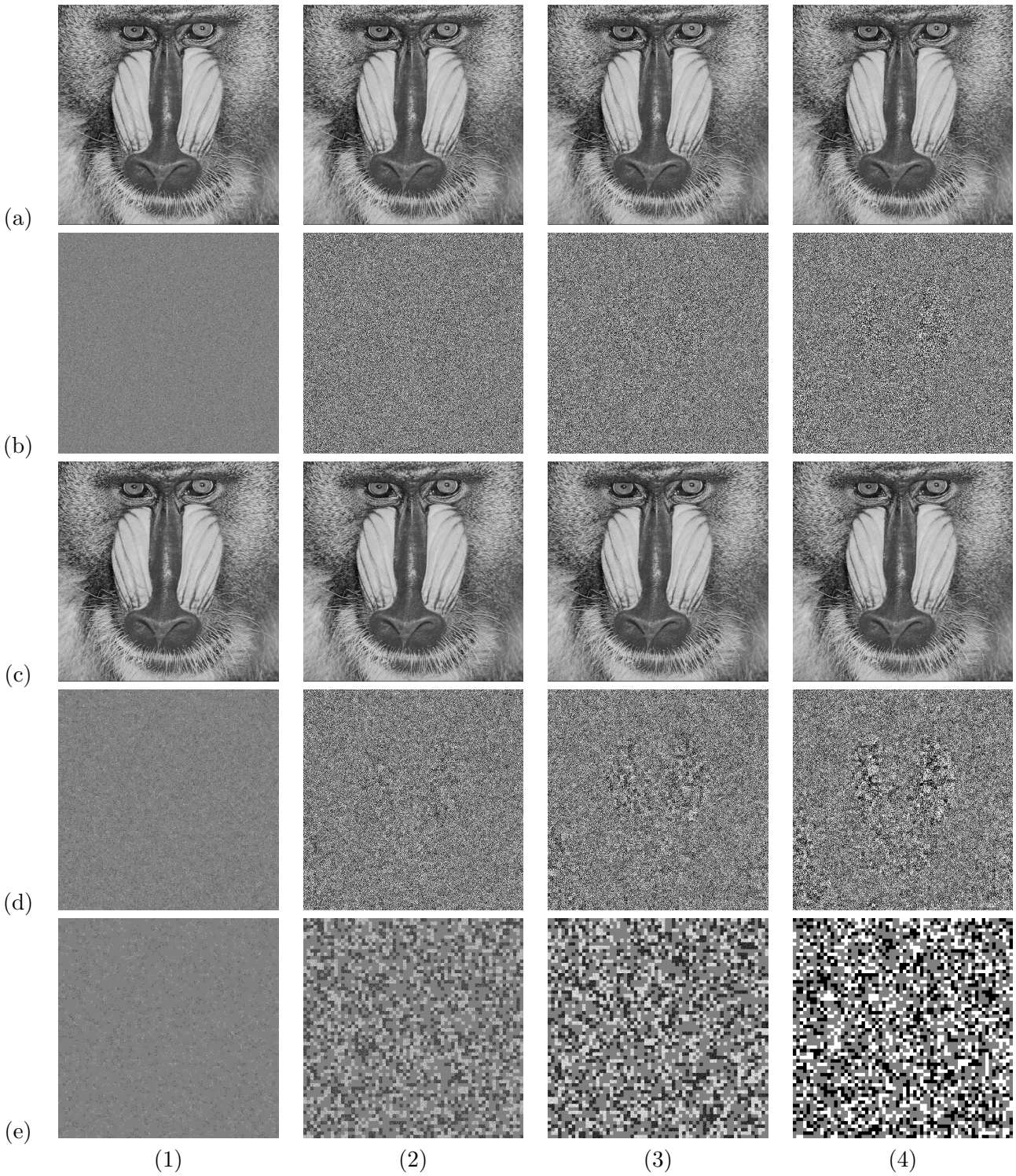


FIG. 40 – *IDC avant quantification, a) Image Baboon comprimée, b) Différence entre l'image originale et l'image comprimée, c) Image Baboon marquée, d) Différence entre l'image originale et l'image marquée, e) Différence entre l'image comprimée et l'image marquée. (1) : $FQ = 100\%$, (2) : $FQ = 80\%$, (3) : $FQ = 70\%$, (4) : $FQ = 50\%$.*

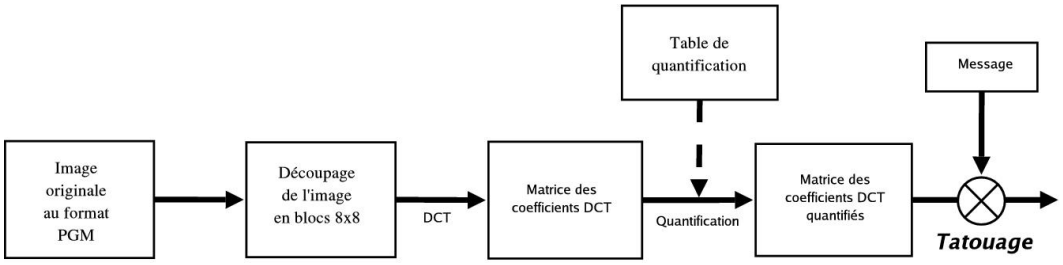


FIG. 41 – *IDC classique, après quantification.*

comme présentée figure 41. Nous venons de voir, section 3.4 que ce n'est pas la seule solution possible, ni forcément la plus judicieuse. En effet la quantification se résume à une division entière, et donc à un arrondi des coefficients DCT. Elle induit une perte de précision. En réalisant l'insertion avant la quantification, il est possible d'intégrer ce reste dans l'IDC et par conséquent d'améliorer la qualité de l'image marquée et comprimée. Dans notre méthode présentée section 3.4, l'IDC se combine avec la phase de quantification mais peut-être remontée en amont du traitement jusque dans le domaine spatial.

3.5.1 Comparaison des IDC classiques et de l'IDC avant quantification

Exemple

- *Bit à insérer,* $b_t = 1$
- *Facteur de quantification du coefficient,* $Q(u,v) = 10$
- *Valeur du coefficient,* $F(u,v) = 96$
- *Valeur du coefficient quantifié,* $F'(u,v) = \left[\frac{96}{10}\right] = 10$

Valeur du coefficient quantifié marqué classiquement, $\mathbf{F'_{dh}(u,v) = 11}$.

En s'intéressant à la valeur avant quantification, il est facile de noter que les deux multiples de $Q(u,v)$ *les plus proches de* $F(u,v)$ *sont 90 et 100. L'IDC va alors consister à choisir parmi ces deux valeurs celle qui donne un bit de poids faible identique au bit à tatouer,* b_t*. Dans notre cas,* $\mathbf{F'_{dh}(u,v) = 9}$ *sera la solution.*

Les deux méthodes ne donnent pas la même solution, l'IDC avant quantification conduit à une estimation plus proche de la valeur flottante du coefficient quantifié, 9.6***.***

Sur cet exemple, l'IDC avant quantification conduit à une variation plus faible du coefficient quantifié. Ce phénomène se généralise.

Une IDC fréquentielle substitue au bit de poids faible du coefficient quantifié un bit du message à insérer[3] :

$$\begin{aligned} F'_{dh_{AQ}}(u,v) &= F'(u,v) - F'(u,v)\%2 + b \\ &= \left[\frac{F(u,v)}{Q(u,v)}\right] - \left[\frac{F(u,v)}{Q(u,v)}\right]\%2 + b. \end{aligned} \tag{49}$$

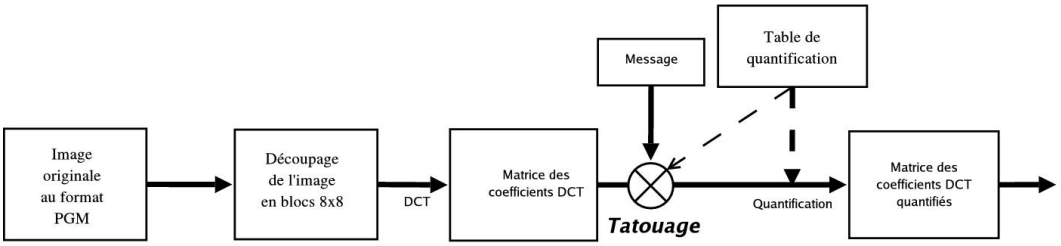


FIG. 42 – *IDC avant quantification.*

Une IDC fréquentielle, avant quantification sur le bit de poids faible, présenté figure 42, conduit à la formule suivante[4] :

$$F_{dh_{BQ}}(u,v) = \begin{cases} \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor Q(u,v) \text{ si } \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor \%2 = b \\ \\ \left\lceil \frac{F(u,v)}{Q(u,v)} \right\rceil Q(u,v) \text{ si } \left\lceil \frac{F(u,v)}{Q(u,v)} \right\rceil \%2 = b \end{cases}. \tag{50}$$

Pour pouvoir comparer ces deux classes d'IDC nous allons exprimer dans les deux cas, la même grandeur, $F'(u,v)$:

$$F'_{dh_{BQ}}(u,v) = \begin{cases} \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor \text{ si } \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor \%2 = b \\ \\ \left\lceil \frac{F(u,v)}{Q(u,v)} \right\rceil \text{ si } \left\lceil \frac{F(u,v)}{Q(u,v)} \right\rceil \%2 = b \end{cases}. \tag{51}$$

Les résultats de la comparaison sont présentés dans le tableau 3.5.1. La différence δ, est la partie flottante de $\frac{F(u,v)}{Q(u,v)}$: $\delta = \frac{F(u,v)}{Q(u,v)} - \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor$. L'IDC avant quantification offre dans tous les cas une variation des coefficients inférieure ou égale à une IDC classique.

3.5.2 Indices objectifs de qualité

Nous allons nous intéresser aux indices objectifs de qualité pour évaluer l'IDC avant quantification par rapport à celui après quantification. La qualité subjective dépend de l'image originale, de ses zones homogènes, des zones texturées et ne peut s'appliquer ici.

3. AQ : After quantization
4. BQ : Before quantization

$\left\lceil \frac{F(u,v)}{Q(u,v)} \right\rceil$	$\left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor$				$\left\lceil \frac{F(u,v)}{Q(u,v)} \right\rceil$			
$\left\lceil \frac{F(u,v)}{Q(u,v)} \right\rceil \%2$	0		1		0		1	
b_t	0	1	0	1	0	1	0	1
$\left\lvert \Delta F'_{dh_{AQ}}(u,v) \right\rvert$	δ	$1-\delta$	$1+\delta$	δ	$1-\delta$	$2-\delta$	δ	$1-\delta$
$\left\lvert \Delta F'_{dh_{BQ}}(u,v) \right\rvert$	δ	$1-\delta$	$1-\delta$	δ	$1-\delta$	δ	δ	$1-\delta$
$\left\lvert \Delta F'_{dh_{AQ}}(u,v) \right\rvert - \left\lvert \Delta F'_{dh_{BQ}}(u,v) \right\rvert$	0	0	2δ	0	0	$2-2\delta$	0	0

TAB. 5 – *Différence de précision entre les deux méthodes en fonction des cas possibles.*

3.5.2.1 Calcul de l'erreur quadratique moyenne fréquentielle

Le premier indice utilisé est l'erreur quadratique moyenne fréquentielle (EQMF), elle se base sur la EQM classique présentée par l'équation (36) et permet d'avoir une idée de l'erreur commise au niveau fréquentiel entre l'image originale et l'image marquée.

Nous travaillons sur N_b blocs de taille 8×8. Nous allons faire une première somme à cette échelle :

$$EQMF = \frac{1}{N_b} \sum_{b=0}^{N_b-1} EQMF_b,$$

puis une seconde pour atteindre l'unité qui nous intéresse, le coefficient DCT :

$$EQMF_b = \frac{1}{64} \sum_{u=0}^{7} \sum_{v=0}^{7} \Delta F(u,v)^2.$$

Nous obtenons ainsi la formule :

$$EQMF = \frac{1}{N} \sum_{b=0}^{N_b-1} \sum_{u=0}^{7} \sum_{v=0}^{7} \Delta F(u,v)^2,$$

avec, $\Delta F(u,v)^2 = Q(u,v)^2 \left(\frac{F(u,v)^2}{Q(u,v)^2} - F'_{dh}(u,v)^2 \right)$.

Nous nous intéressons à la différence de qualité, non pas entre les deux images marquées, mais entre chacune d'elles et l'image originale.

$$\begin{aligned}\Delta_{dh_{BQ}}^{dh_{AQ}} EQMF &= EQMF_{dh_{AQ}} - EQMF_{dh_{BQ}} \\ &= \frac{1}{N}\sum_{b=0}^{N_b-1}\sum_{u=0}^{7}\sum_{v=0}^{7} Q(u,v)^2 \left(\Delta_{dh_{AQ}} F'_b(u,v)^2 - \Delta_{dh_{BQ}} F'_b(u,v)^2\right). \quad (52)\end{aligned}$$

La EQMF n'est pas forcément un très bon indicateur pour juger de la qualité objective d'une image car nous ne considérons que les variations des coefficients DCT. Ceux-ci n'ont pas tous la même influence sur l'image et ils peuvent interférer entre eux.

3.5.2.2 Calcul de l'erreur quadratique moyenne

L'EQM est un indicateur plus parlant car il se place dans le domaine spatial. Il donne une idée des modifications apportées à l'image.

Nous conservons la première décomposition de l'image en blocs 8×8, puisque les variations sur les pixels découlent directement des coefficients DCT de chaque bloc :

$$\begin{aligned}EQM &= \frac{1}{N_b}\sum_{b=0}^{N_b-1} EQM_b \\ EQM_b &= \frac{1}{64}\sum_{i=0}^{7}\sum_{j=0}^{7} \Delta p_b(i,j)^2 \\ EQM &= \frac{1}{N}\sum_{b=0}^{N_b-1}\sum_{i=0}^{7}\sum_{j=0}^{7} \Delta p(i,j)^2,\end{aligned}$$

avec :

$$\Delta p(i,j) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)Q(u,v)\Delta F(u,v)\cos\left(\frac{\Pi(2i+1)u}{16}\right)\cos\left(\frac{\Pi(2j+1)v}{16}\right).$$

Nous nous intéressons aux variations de la EQM :

$$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM = EQM_{dh_{AQ}} - EQM_{dh_{BQ}} = \frac{1}{16N}\sum_{b=0}^{N_b-1}\sum_{i=0}^{7}\sum_{j=0}^{7}$$
$$\left(\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)Q(u,v)\Delta_{dh_{AQ}} F'_b(u,v)\cos\left(\frac{\Pi(2i+1)u}{16}\right)\cos\left(\frac{\Pi(2j+1)v}{16}\right)\right)^2$$
$$-\left(\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)Q(u,v)\Delta_{dh_{BQ}} F'_b(u,v)\cos\left(\frac{\Pi(2i+1)u}{16}\right)\cos\left(\frac{\Pi(2j+1)v}{16}\right)\right)^2. \quad (53)$$

Cette équation est difficile à réduire du fait des termes croisés. Si nous ne considérons la variation que d'un seul coefficient DCT, $F(u_k,v_k)$, elle se simplifie :

$$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM = \frac{1}{16N} \sum_{b=0}^{N_b-1} \sum_{i=0}^{7} \sum_{j=0}^{7} C(u_k)^2 C(v_k)^2 Q(u_k,v_k)^2 \times$$
$$\left(\Delta_{dh_{AQ}} F'_b(u_k,v_k) - \Delta_{dh_{BQ}} F'_b(u_k,v_k)\right)^2 \cos\left(\frac{\Pi(2i+1)u_k}{16}\right)^2 \cos\left(\frac{\Pi(2j+1)v_k}{16}\right)^2. \quad (54)$$

Le résultat suivant permet une simplification importante :

$$\frac{1}{16} C(u_k)^2 C(v_k)^2 \sum_{i=0}^{7} \sum_{j=0}^{7} \cos\left(\frac{\Pi(2i+1)u_k}{16}\right)^2 \cos\left(\frac{\Pi(2j+1)v_k}{16}\right)^2 = 1.$$

Nous obtenons finalement :

$$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM = \frac{Q(u_k,v_k)^2}{N} \sum_{b=0}^{N_b-1} \Delta_{dh_{AQ}} F'_b(u_k,v_k)^2 - \Delta_{dh_{BQ}} F'_b(u_k,v_k)^2. \quad (55)$$

3.5.2.3 Pic du rapport signal à bruit (PSNR)

Là encore, nous nous intéressons aux variations de cet indicateur entre les deux tatouages. A partir de l'équation (37), nous avons :

$$\begin{aligned} \Delta_{dh_{AQ}}^{dh_{BQ}} PSNR &= 10log_{10} \frac{255^2}{EQM_{dh_{BQ}}} - 10log_{10} \frac{255^2}{EQM_{dh_{AQ}}} \\ &= 10log_{10} \frac{EQM_{dh_{AQ}}}{EQM_{dh_{BQ}}} \\ &= 10log_{10} \frac{EQM_{dh_{BQ}} + \Delta_{dh_{BQ}}^{dh_{AQ}} EQM}{EQM_{dh_{BQ}}}. \end{aligned} \quad (56)$$

3.5.3 Applications aux méthodes de tatouage robustes à la compression JPEG

Dans cette section, nous proposons de modifier les méthodes existantes présentées section 3.3 en utilisant une IDC avant quantification. En effet, notre méthode peut s'appliquer à tous les algorithmes d'IDC de ce type s'appuyant sur le bit de poids faible uniquement. L'erreur d'arrondi commise par l'insertion après quantification est fonction de la partie flottante et ne peut dépasser 1.

Nous allons comparer les deux méthodes pour le tatouage sur la composante DC, le JPEG-JSTEG et une version restreinte au LSB du tatouage par modification de la table de quantification. Nous nous appuierons sur les facteurs de qualité 50% qui utilise la matrice

de quantification standard présentée figure 40 et 100% qui ne quantifie pas les coefficients DCT (tous les coefficients de quantification sont à 1).

3.5.3.1 Facteur de qualité de 50%

3.5.3.1.1 Estimation pour les résultats théoriques Nous avons besoin de faire une estimation sur la répartition des différents événements intervenants pour obtenir des résultats théoriques. Néanmoins, ces derniers ne seront pas toujours proposés, soit du fait de la complexité du calcul de la EQM quand le tatouage porte sur plusieurs termes, soit du fait de l'impossibilité de proposer des répartitions satisfaisantes pour les différents événements intervenants. C'est notamment le cas pour un tatouage sur tous les coefficients et un facteur de qualité de 50% : les répartitions ne seront pas les mêmes à cause des différences de quantification pour chacun des coefficients et il sera difficile d'en proposer une globale.

Dans le cas où des résultats théoriques sont présentés, il s'appuie sur la modélisation suivante :

- équiprobabilité pour la valeur du bit à tatouer :

$$p(b_t = 1) = p(b_t = 0) = \frac{1}{2}.$$

- équiprobabilité de la valeur initiale du LSB de la composante continue :

$$p\left(\left[\frac{F(u,v)}{Q(u,v)}\right] \%2 = 0\right) = p\left(\left[\frac{F(u,v)}{Q(u,v)}\right] \%2 = 1\right) = \frac{1}{2}.$$

- équiprobabilité de la partie flottante de $\frac{F(u,v)}{Q(u,v)}$ sur l'intervalle [0,1[que nous réduisons aux intervalles [0,0.5[, [0.5,1[:

$$p(\delta \in [0,0.5[) = p(\delta \in [0.5,1[) = \frac{1}{2}.$$

Ainsi :

$$\begin{aligned}\sum_{b=0}^{N_b-1} \Delta F'_{dh_{AQ}}(0,0)^2 - \Delta F'_{dh_{BQ}}(0,0)^2 &= \frac{N_b}{8}((1+0.25)^2 - (1-0.25)^2 + (2-0.75)^2 - 0.75^2) \\ &= \frac{N_b}{4}\end{aligned}$$

$$\begin{aligned}\Delta^{dh_{AQ}}_{dh_{BQ}} EQMF &= \frac{N_b Q(0,0)^2}{4N} \\ &= \frac{Q(0,0)^2}{256} \\ \Delta^{dh_{AQ}}_{dh_{BQ}} EQM &= \frac{Q(0,0)^2}{256},\end{aligned}$$

donc :

$$\Delta^{dh_{BQ}}_{dh_{AQ}} PSNR = 10log_{10}\frac{EQM_{dh_{BQ}} + \frac{Q(0,0)^2}{256}}{EQM_{dh_{BQ}}}.$$

	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQMF$	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM$	$\Delta_{dh_{AQ}}^{dh_{BQ}} PSNR$	Gain relatif sur le PSNR
théorie	1	1		
lena	1.031	1.029	0.179	0.5%
baboon	0.972	0.970	0.036	0.1%
boat	1.002	1	0.185	0.5%
girl	1.003	1.006	0.228	0.6%

TAB. 6 – *Variation des indicateurs de qualité pour une IDC sur la composante DC et un facteur de qualité de 50%.*

3.5.3.1.2 IDC sur la composante DC La qualité des images produites est voisine de 30 dB. Le gain qu'apporte une IDC avant quantification donné par le tableau 6, environ 0.5%, reste marginal. L'image Baboon présente des résultats en accord avec la théorie pour les indicateurs où celle-ci est disponible, mais en désaccord avec la tendance des autres images pour le reste.

3.5.3.1.3 IDC selon la méthode JPEG-JSTEG Le tableau 7 montre des résultats sensiblement identiques aux précédents : un gain relatif inférieur à 1% et l'image Baboon qui ne suit pas la tendance générale. Ces ressemblances ne sont pas étonnantes : avec un facteur de qualité de 50%, les composantes quantifiées répondant aux critères du JPEG-JSTEG se limitent à quelques unes localisées dans les basses fréquences.

	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQMF$	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM$	$\Delta_{dh_{AQ}}^{dh_{BQ}} PSNR$	Gain relatif sur le PSNR
lena	7.657	1.499	0.226	0.7%
baboon	27.1	0.111	0.004	0.01%
boat	9.506	1.818	0.275	0.8%
girl	7.081	1.462	0.280	0.8%

TAB. 7 – *Variation des indicateurs de qualité pour une IDC par la méthode JPEG-JSTEG et un facteur de qualité de 50%.*

3.5.3.1.4 IDC par modification de la matrice de quantification Le tableau 8 présente un gain plus faible pour l'IDC par modification de la matrice de quantification. L'insertion est effectuée sur des composantes non quantifiées et la variation induite par le choix de l'effectuer avant ou après la quantification est faible comparativement aux

dégradations apportées par celle-ci sur les autres coefficients. La qualité des images obtenues reste correcte, juste inférieure à 40 dB.

	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQMF$	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM$	$\Delta_{dh_{AQ}}^{dh_{BQ}} PSNR$	Gain relatif sur le PSNR
lena	0.408	0.099	0.038	0.1%
baboon	0.406	0.129	0.007	0.02%
boat	0.410	0.096	0.049	0.1%
girl	0.405	0.110	0.058	0.1%

TAB. 8 – *Variation des indicateurs de qualité pour une IDC par modification de la matrice de quantification et un facteur de qualité de 50%.*

3.5.3.1.5 IDC sur tous les coefficients DCT L'IDC sur tous les coefficients offre un gain flagrant. Néanmoins, les images finales sont fortement dégradées, leur qualité avoisine 15 dB. De plus, les variations sur l'image sont tellement importantes que des débordements sont très probables. Le seuillage des intensités aux valeurs extrêmes du spectre des niveaux de gris fausse alors les résultats. Avec un facteur de qualité de 50%, le gain ne se révèle pas flagrant sauf pour une IDC sur toutes les composantes. Il présente un faible intérêt vu les dégradations subies par l'image.

	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQMF$	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM$	$\Delta_{dh_{AQ}}^{dh_{BQ}} PSNR$	Gain relatif sur le PSNR
lena	192.498	192.632	0.384	2.6%
baboon	511.741	511.824	1.059	7%
boat	161.579	162.08	0.321	2.1%
girl	144.973	145.314	0.288	1.9%

TAB. 9 – *Variation des indicateurs de qualité pour une IDC sur tous les coefficients DCT et un facteur de qualité de 50%.*

3.5.3.2 Facteur de qualité de 100%

Un facteur de qualité de 100% ne quantifie pas les coefficients DCT :

$$F'(u,v) = [F(u,v)] .$$

Ainsi, les estimations utilisées auparavant dans certains cas seulement sont valables maintenant partout. Les seules restrictions pour la présentation de valeurs théoriques

proviennent de la complexité de l'évaluation de l'EQM dans le cas d'une IDC sur plusieurs coefficients.

3.5.3.2.1 IDC sur la composante DC Le tableau 10 montre un gain identique pour une IDC sur la composante DC entre des facteurs de qualité de 50% et 100%. La différence de qualité est toutefois sensible : la qualité des images est ici de 60 dB. Pour la variation de la EQM, il y a un décalage très constant entre la théorie et les résultats pratiques.

	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQMF$	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM$	$\Delta_{dh_{AQ}}^{dh_{BQ}} PSNR$	Gain relatif sur le PSNR
théorie	0.004	0.004		
lena	0.004	0.007	0.335	0.6%
baboon	0.004	0.008	0.368	0.6%
boat	0.004	0.008	0.369	0.6%
girl	0.004	0.007	0.364	0.6%

TAB. 10 – *Variation des indicateurs de qualité pour une IDC sur la composante DC et un facteur de qualité de 100%.*

3.5.3.2.2 IDC selon la méthode JPEG-JSTEG Le tableau 11 présente un gain important apporté par le choix d'une IDC avant quantification. De plus la qualité des images est très bonne, environ 50 dB.

	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQMF$	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM$	$\Delta_{dh_{AQ}}^{dh_{BQ}} PSNR$	Gain relatif sur le PSNR
lena	0.508	0.157	1.55	3%
baboon	0.603	0.215	1.892	3.8%
boat	0.443	0.122	1.321	2.6%
girl	0.430	0.122	1.355	2.6%

TAB. 11 – *Variation des indicateurs de qualité pour une IDC selon la méthode JPEG-JSTEG et un facteur de qualité de 100%.*

3.5.3.2.3 IDC par modification de la matrice de quantification Les résultats du tableau 12 sont voisins de ceux du précédent : un gain conséquent et une bonne conservation de la qualité de l'image (52dB). Sans quantification, toutes les composantes se comportent de la même manière et le JPEG-JSTEG peut s'appuyer sur bon nombre d'entre elles. Il est néanmoins rare qu'une image est besoin de toutes les fréquences pour être représentée,

ce qui rapproche le nombre de coefficients marqués et conduit aux fortes ressemblances entre les deux méthodes pour ce facteur de qualité.

	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQMF$	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM$	$\Delta_{dh_{AQ}}^{dh_{BQ}} PSNR$	Gain relatif sur le PSNR
lena	0.356	0.109	1.598	3%
baboon	0.357	0.109	1.582	3%
boat	0.357	0.108	1.565	3%
girl	0.356	0.112	1.636	3.1%

TAB. 12 – *Variation des indicateurs de qualité pour une IDC par modification de la matrice de quantification et un facteur de qualité de 100%.*

3.5.3.2.4 IDC sur tous les coefficients DCT L'IDC sur toutes les composantes présente les résultats les plus probants, tableau 13 : une qualité d'image toujours très bonne, autour de 52 dB et un gain relatif de 4%, équivalent en gain absolu à 2 dB, quantité loin d'être négligeable.

	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQMF$	$\Delta_{dh_{BQ}}^{dh_{AQ}} EQM$	$\Delta_{dh_{AQ}}^{dh_{BQ}} PSNR$	Gain relatif sur le PSNR
théorie	0.250			
lena	0.251	0.251	2.057	3.9%
baboon	0.251	0.250	2.046	3.9%
boat	0.249	0.248	2.028	3.9%
girl	0.249	0.248	2.040	3.9%

TAB. 13 – *Variation des indicateurs de qualité pour une IDC sur tous les coefficients DCT et un facteur de qualité de 100%.*

FIG. 43 – *Image originale.*

Le gain qualitatif par une IDC avant quantification est très visible pour un facteur de qualité de 100%, il devient négligeable quand ce facteur descend à 50%. Avec une telle

PSNR = 34.079 dB (a) PSNR = 34.258 dB (b) PSNR = 58.228 dB (c) PSNR = 58.563 dB (d)

FIG. 44 – *IDC sur la composante DC. (a) classique (après quantification), FQ=50%; (b) avant quantification, FQ=50%; (c) classique (après quantification), FQ=100%; (d) avant quantification, FQ=100%.*

PSNR = 33.420 dB (a) PSNR = 33.646 dB (b) PSNR = 50.940 dB (c) PSNR = 52.491dB (d)

FIG. 45 – *IDC selon la méthode JPEG-JSTEG. (a) classique (après quantification), FQ=50%; (b) avant quantification, FQ=50%; (c) classique (après quantification), FQ=100%; (d) avant quantification, FQ=100%.*

PSNR = 37.608 dB (a) PSNR = 37.647 dB (b) PSNR = 52.614 dB (c) PSNR = 54.204 dB (d)

FIG. 46 – *IDC par modification de la matrice de quantification. (a) classique (après quantification), FQ=50%; (b) avant quantification, FQ=50%; (c) classique (après quantification), FQ=100%; (d) avant quantification, FQ=100%.*

PSNR = 14.563 dB (a) | PSNR = 14.948 dB (b) | PSNR = 49.894 dB (c) | PSNR = 51.951 dB (d)

FIG. 47 – *IDC sur toutes les composantes. (a) classique (après quantification), FQ=50%; (b) avant quantification, FQ=50%; (c) classique (après quantification), FQ=100%; (d) avant quantification, FQ=100%.*

compression, l'oeil humain détecte les dégradations sur l'image. Ainsi, prendre la peine de réaliser l'insertion sur le LSB des composantes fréquentielles avant quantification apparaît très intéressant.

Les résultats visuels sont présentés figure 43 à figure 47 pour l'image Lena. A l'oeil, les différences ne sont pas forcément perceptibles, mais l'évaluation du PSNR en donne une idée. L'IDC sur toutes les composantes présente des perturbations importantes et régulières dues aux débordements des intensités hors du spectre des niveaux de gris. Les valeurs d'intensités obtenues à partir des coefficients DCT ne sont pas ramenées à l'intérieur du spectre et par effet de boucle correspondent à des valeurs à l'autre extrémité de celui-ci.

3.6 Insertion des données basées contenu

Dans les sections précédentes, l'IDC concerne des images sans se soucier du contenu. Mais l'information peut se rapporter aux objets de celle-ci. Il convient alors de la localiser [Lovarco 04b]. L'IDC contextuelle en est une supplémentaire [Bas 02]. Sa particularité est que le message est relatif à l'image.

Cette nouvelle catégorie d'IDC a des applications évidentes pour l'indexation dans une base de données d'images. Les utilisateurs peuvent extraire des images de la base, les manipuler et intégrer les nouvelles images obtenues. Les images seront définies par leur contenu à partir des objets qui les composent. L'indexation devient simple si ces images contiennent cette information sous forme d'une IDC robuste et localisée.

Les méthodes d'IDC nécessitent une robustesse contre des modifications modérées dues à des traitements classiques tels que la compression, le découpage ou les rotations. Dans

cette section nous présentons une méthode qui insère un grand nombre de bits dans chaque région d'intérêt (RI) de l'image couleur. Le nombre de bits est fonction de la taille de la RI. Cette méthode est une extension aux images couleurs [Chareyron 02] de la méthode développée section 3.4. Cette méthode s'appuie donc également sur les composantes DC de la DCT [Bors 98, Cox 97a, Koch 95].

Les données cachées sont synchronisées avec les RIs. Cette méthode résiste principalement aux manipulations géométriques [Kutter 98, Queluz 99, Ruanaidh 98] et à la compression JPEG. Dans la section 3.6.1 nous décrivons l'utilisation de l'analyse en composante principale (ACP) afin de pouvoir s'appuyer sur le contenu de l'image. Dans la section 3.6.2 nous détaillons l'extension de notre méthode aux images couleurs, et enfin, dans la section 3.6.3, nous présentons des résultats de cette nouvelle méthode.

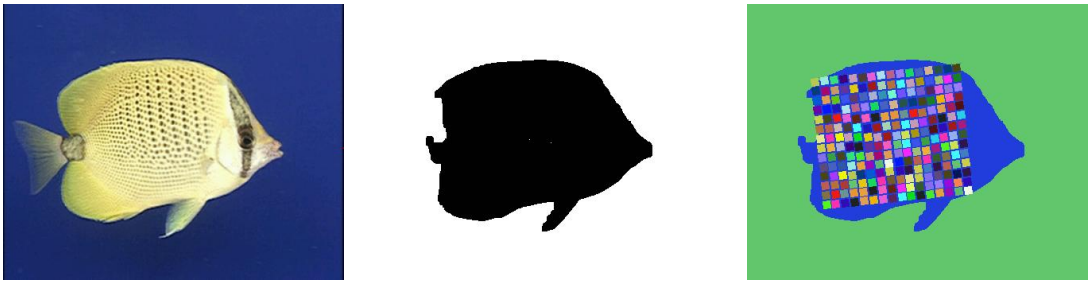
3.6.1 Extraction et description des régions d'intérêt

3.6.1.1 Obtention d'un masque de l'image

Notre objectif n'est pas de découper l'image en RI à partir d'un algorithme simple et stable de segmentation [Healey 89]. Le contenu souhaité de l'image doit être décrit par des RIs non connectées. Ces RIs doivent avoir une taille minimale pour être considérée par l'IDC. Nous retenons uniquement les régions d'une taille supérieure à 3000 pixels. Pour obtenir le masque de l'image nous convertissons l'image couleur en niveau de gris et nous utilisons un algorithme de croissance de régions. Pour chaque pixel, tous les pixels voisins sont rajoutés à la même région si leur niveau de gris est similaire, sinon ils constituent la base d'une autre région. La valeur du seuil doit être réglée par l'utilisateur mais peut également faire partie de la clef. Chaque RI est donc un ensemble de pixels décrivant grossièrement les formes présentes dans l'image support. Pour illustrer ce procédé, une segmentation est appliquée sur l'image originale nommée Fish, figure 48.a, et nous obtenons le masque associé illustré figure 48.b.

Après segmentation, l'image est un ensemble de zones convexes. Chaque région dispose d'un contour composé de pixels appelés pixels frontières. Ces pixels seront les premiers à changer de RI lors d'une transformation géométrique. Par conséquent, afin d'augmenter la robustesse face à ces transformations, les pixels frontières ne sont pas considérés comme appartenant à la RI. Afin de régulariser les contours des RIs, un ensemble d'érosions et de dilatations sont appliquées sur les RIs [Serra 88].

Après réduction des RIs un étiquetage en composantes connexes peut-être appliqué. L'étiquetage en composantes connexes d'une image binaire consiste à attribuer une étiquette



(a) (b) (c)

FIG. 48 – *a) L'image originale "Fish", b) Le masque binaire associé, c) Forme et orientation des blocs dans la RI.*

différente pour chaque composante connexe et identique pour tous les pixels d'une même composante. Ce traitement utilisé s'appuie sur l'opérateur de Rosenfeld [Rosenfeld 68]. La structure de l'algorithme utilisé impose un travail en 4-connexité. Finalement chaque pixel prend comme étiquette une valeur L_{ROI} spécifique à sa RI d'appartenance.

3.6.1.2 Caractéristiques des RIs

Après extraction des RIs, une analyse de celles-ci est nécessaire afin de pouvoir insérer les données. En effet, une caractérisation précise de ces RI nous permettra alors, après des déformations géométriques, d'extraire les données cachées. Les données cachées doivent donc être insérées à partir de repères dépendant des propriétés de chaque RI qui sont : la taille, la position et l'orientation des RIs.

– Un descripteur de taille, la surface de la RI notée $S(RI)$, est le premier paramètre calculé. Il correspond au nombre de pixels qui ont la même étiquette L_{RI}. Pour faciliter le traitement de ces RIs une normalisation est effectuée. Cette normalisation est d'une taille moyenne de RIs notée S_s. Un facteur de taille est finalement obtenu :

$$F_S(RI) = \frac{S(RI)}{S_s}. \tag{57}$$

S_s est généralement égale à 10% de la taille de l'image originale, c'est à dire 15000 pixels pour une image comportant 400×350 pixels comme l'image Fish. L'utilisation de ce changement de taille rend possible la détection des données après un changement d'échelle de l'image.

– Le barycentre G_{RI} des RIs est utilisé comme descripteur de position. En effet, les moments du premier degré de la RI, notés $\mu_i(RI)$ et $\mu_j(RI)$ localisent précisément ce

point particulier. Les paramètres $\mu_\imath(ROI)$ et $\mu_\jmath(ROI)$ correspondent respectivement à la moyenne des abscisses et des ordonnées de la RI :

$$\begin{cases} \mu_\imath(RI) = \dfrac{1}{S(RI)} \displaystyle\sum_{k=0}^{S(RI)-1} \imath(k) \\ \\ \mu_\jmath(RI) = \dfrac{1}{S(RI)} \displaystyle\sum_{k=0}^{S(RI)-1} \jmath(k), \end{cases} \tag{58}$$

où $\imath(k)$ et $\jmath(k)$ sont respectivement les coordonnées verticale et horizontale du pixel k.

Nous obtenons finalement :

$$G_{RI} = G[\mu_\imath(RI),\mu_\jmath(RI)]. \tag{59}$$

Le barycentre G_{RI} constitue le point d'origine du référentiel rattaché à la RI. Ce repère détermine le point de départ de l'insertion et de la détection. C'est lui qui permet la synchronisation des données insérées dans l'image.

– Le descripteur de forme et de direction est basé sur l'analyse en composantes principales (ACP). Notre schéma d'insertion utilise des repères qui dépendent de la forme des RIs. Pour construire chacun de ces référentiels, une origine et deux directions sont nécessaires. La position du point G_{ROI} a été calculée précédemment. Pour calculer les directions spécifiques de chacune des RIs nous avons utilisé une méthode dérivée de l'ACP qui implique le calcul des moments du second degré des RIs. Nous obtenons les variances horizontale et verticale, $V_x(RI)$ et $V_y(RI)$ ainsi que la covariance $V_{xy}(RI)$:

$$\begin{cases} V_x(RI) &= \dfrac{1}{S(RI)} \displaystyle\sum_{k=0}^{S(RI)-1} (\imath(k) - \mu_\imath(RI))^2 \\ \\ V_y(RI) &= \dfrac{1}{S(RI)} \displaystyle\sum_{k=0}^{S(RI)-1} (\jmath(k) - \mu_\jmath(RI))^2 \\ \\ V_{xy}(RI) &= \dfrac{1}{S(RI)} \displaystyle\sum_{k=0}^{S(RI)-1} [\imath(k) - \mu_\imath(RI)][\jmath(k) - \mu_\jmath(RI)]. \end{cases} \tag{60}$$

L'ensemble de ces coefficients peut être représenté par une matrice $C_0(RI)$ de 2×2 éléments :

$$C_0(RI) = \begin{bmatrix} V_x(RI) & V_{xy}(RI) \\ V_{xy}(RI) & V_y(RI) \end{bmatrix}. \tag{61}$$

Comme le montre le système d'équations (60), les variances spatiales donnent des indications à la fois de forme et de direction sur les RIs. Par exemple, une RI avec une variance importante en x et une faible variance en y sera allongée selon l'axe des abscisses. L'analyse de la matrice $C_0(RI)$ nous permet d'extraire deux valeurs propres $\lambda_1(RI)$ et $\lambda_2(RI)$ et deux vecteurs propres $\vec{V_1}(RI)$ et $\vec{V_2}(RI)$ [Bas 01]. Les doublets $\{\lambda_i,\vec{V_i}\}$, avec $i \in \{1,2\}$, représentent les axes majeur et mineur propres à la RI. Ces deux axes sont illustrés figure 49 sur le masque de l'image Fish. Associés au barycentre G_{RI}, ils forment le repère apte à recevoir les données à insérer. L'information à insérer peut alors être synchronisée avec chaque RI de l'image.

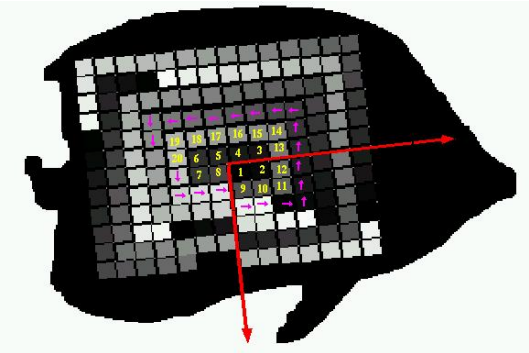


FIG. 49 – *Repère de référence et ordre de construction des blocs dans la RI.*

3.6.1.3 Synchronisation des données cachées avec les RIs

Notre méthode d'IDC utilise des bloc unitaires de n^2 pixels. Ces blocs sont définis différemment selon la forme et l'orientation des RIs. De plus, ils sont construits suivant un ordre précis afin de préserver l'intégrité de l'information insérée. La taille des blocs est normalisée afin de rendre la détection unique et rapide.

Il faut ensuite enchaîner des blocs afin de pouvoir insérer une séquence binaire. En effet pour insérer le message un ordre doit être respecté. Nous utilisons également les doublets $\{\lambda_i,\vec{V_i}\}$ associés aux barycentres G_{RI} pour déterminer un chemin d'insertion noté Ip_{ROI}. Ip_{ROI} est un ensemble ordonné de m pixels originels sp_k, avec $0 \leq k < m$ et m le nombre de bits insérés dans la RI. Chaque point sp_k constitue le point de départ d'un bloc unitaire. Nous obtenons ainsi un ensemble de m blocs, illustré Figure 49. Le sens de propagation des blocs suit un développement de type excentrique.

3.6.2 Utilisation d'image couleur pour l'IDC

L'algorithme JPEG transforme une image couleur RGB en une image dans l'espace YCrCb. Dans cet espace couleur, l'essentiel de l'information est contenu dans la plan de luminance Y. L'algorithme JPEG ré-échantillonne les plans de chrominance Cr et Cb afin de comprimer plus fortement une image couleur. Une image couleur en RGB peut être transformée dans l'espace YCrCb suivant les équations suivantes :

$$\begin{cases} \mathbf{Y} &= (0.2989 \times \mathbf{R}) &+ (0.5866 \times \mathbf{G}) &+ (0.1145 \times \mathbf{B}) \\ \mathbf{Cr} &= (0.7132 \times \mathbf{R}) &- (0.7132 \times \mathbf{Y}) &+ 128 \\ \mathbf{Cb} &= (0.5647 \times \mathbf{B}) &- (0.5647 \times \mathbf{Y}) &+ 128. \end{cases} \quad (62)$$

Pour résister à ce type de conversion, nous avons choisi d'insérer l'information cachée dans les trois composantes Y, Cr et Cb.

3.6.3 Résultats

Dans cette section nous appliquons la méthode proposée à deux images contenant des RIs facilement détectables, l'image Objets (1013×760) figure 50.a et l'image Fish (429×347) figure 51.a. Plusieurs traitements sont testés sur ces deux images.

RI	Longueur du message (bits)	Nombre de bits insérés par composantes	taille des blocs (pixels)	taille RI (pixels)	Taux d'insertion
1	28	56	113	10581	59,8%
2	60	120	117	26441	53,1%
3	102	204	113	35261	65,4%
4	18	36	113	7833	51,2%
5	8	16	106	4138	40,9%
6	32	64	113	8620	83,9%
7	96	192	113	68474	31,7%
8	50	100	113	24940	45,3%

TAB. 14 – *Messages insérés et taux d'insertion pour chaque RI.*

A partir de l'image originale Objets figure 50.a, nous avons détecté 8 RIs dans lesquelles un message peut être inséré. L'image est ensuite analysée afin d'obtenir ses caractéristiques telles que les directions principales. La figure 50.b montre la forme des RIs et la construction des blocs marqués. Une étiquette particulière est associée à chaque RI.

Après cette analyse, la méthode d'IDC peut commencer. Afin d'augmenter la robustesse, nous avons choisi d'insérer plusieurs fois les messages. La redondance est donc double.

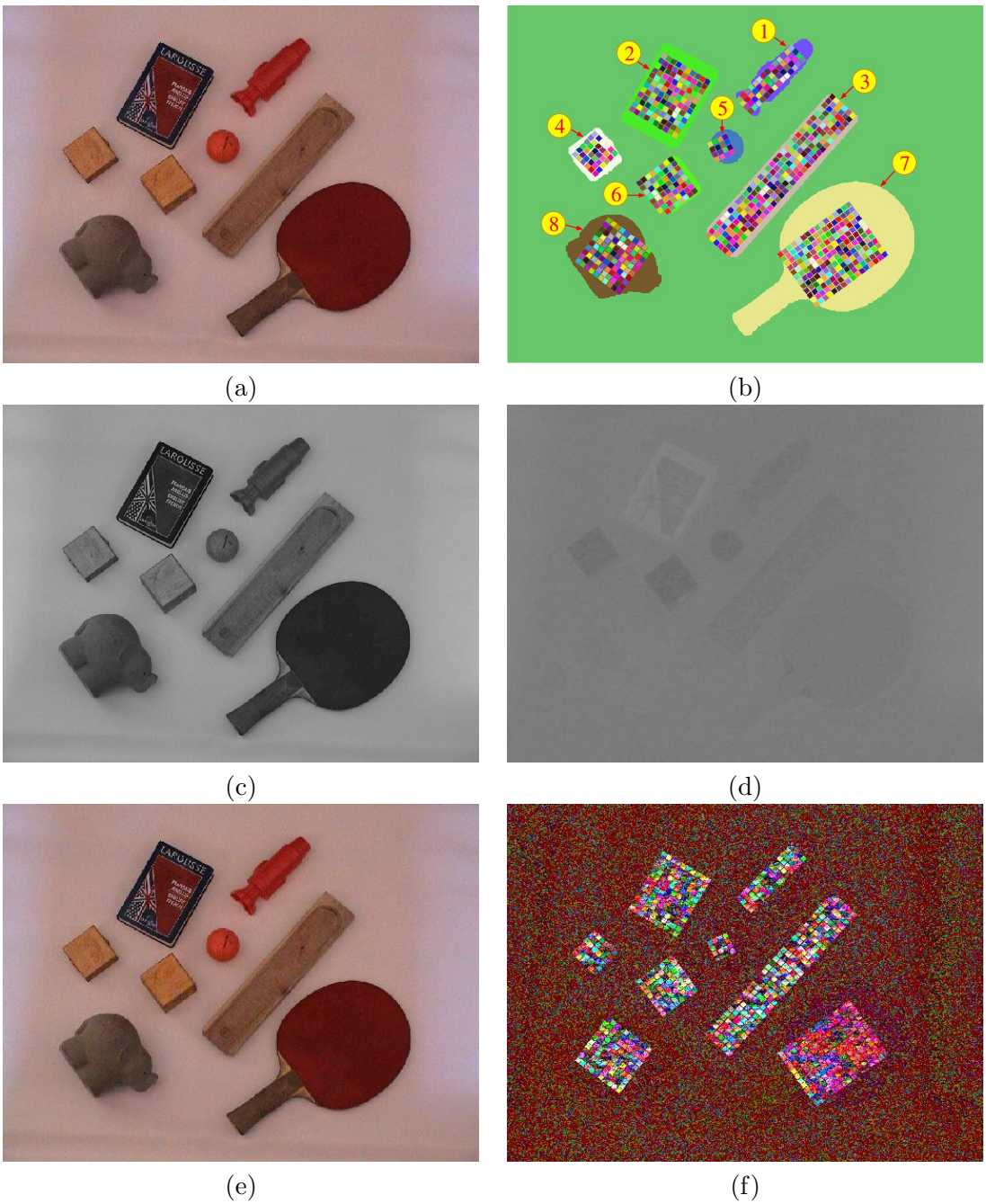


FIG. 50 – *a) Image originale, b) Détection du chemin et des blocs marqués, c) Composante Y marquée, d) Composante Cr marquée, e) Image couleur marquée, f) Différence entre l'image originale et l'image marquée.*

Premièrement, les bits du message sont répétés deux fois dans chaque RI. Ensuite l'information couleur est utilisée pour insérer l'information trois fois : une fois dans chaque composante couleur. Cette redondance de l'information permet à l'extraction d'appliquer un vote majoritaire pour corriger d'éventuelles erreurs. Les figures 50.c et 50.d présentent respectivement les composantes marquées Y et Cr. Finalement l'image couleur marquée est construite avec les trois composantes marquées. Les données embarquées sont invisibles à l'oeil, figure 50.e. Mais si nous effectuons la différence pixel à pixel entre l'image originale Figure 50.a et l'image marquée Figure 50.e nous visualisons les blocs marqués, figure 50.f. Le PSNR entre ces deux images est égal à 50.52 dB.

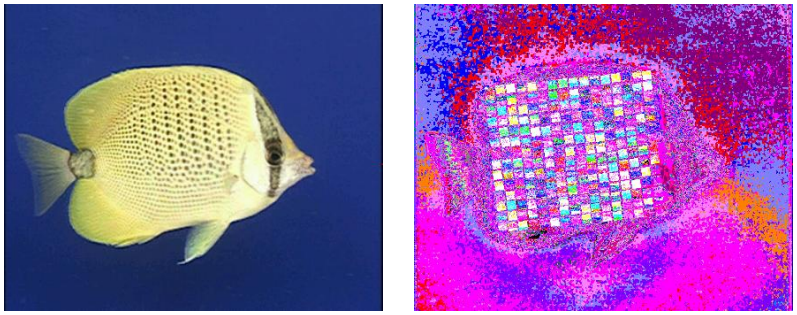
Le tableau 14 contient des informations concernant les données cachées dans chaque RI : la longueur du message, les nombres de bits insérés par composantes, la taille des blocs utilisés, la taille des RIs et le facteur d'insertion. Une première analyse montre que la longueur du message dépend de la taille de la RI. En effet si la taille de la RI augmente alors le nombre de blocs de pixels augmente également. Par conséquence plus de bits du message peuvent être insérés. Par exemple, dans la première RI, 28 bits du message sont embarqués et dans la troisième RI, trois fois plus grande, le nombre de bits embarqués du message est multiplié par 4. Nous observons alors que le nombre de bits détectés est deux fois plus grand que la longueur du message. Cela correspond au premier degré de redondance. Par exemple, dans la cinquième RI, le message binaire est 00110101, et celui détecté est **00110101**00110101. Chaque bit est répété. Par conséquent la qualité de la détection est améliorée. D'un autre coté, la quatrième colonne du tableau 14 présente les tailles des blocs utilisées dans chaque RI. Nous observons que la taille diminue un peu entre chaque RI d'à peu près 6%. Comme notre méthode s'appuie sur des blocs carrés, la donnée d'entrée est la taille d'un bloc aligné avec le repère de l'image. Ensuite la taille des blocs est modifiée en fonction de l'orientation de chacune des RIs. Finalement chaque région a des blocs avec une forme particulière. En utilisant la taille des blocs et la taille des RIs, nous pouvons évaluer le taux d'insertion dans chacune des RIs :

$$E_r(RI) \quad = \quad \frac{taille\ blocs\ RI \ \times \ Nombre\ de\ bits\ insérés\ dans\ la\ RI}{Taille\ RI}. \qquad (63)$$

Ce taux varie entre 31,7% et 83,9% en fonction de la forme de la RI. Ceci montre l'inconvénient d'utiliser des blocs carrés.

Afin de valider plus précisément notre méthode, une autre image a été utilisée pour insérer des données cachées. La figure 51.a montre l'image couleur marquée. La longueur du message est 112 bits et le nombre de bits insérés est 224. Pour visualiser les données

insérées, la différence pixel à pixel entre l'image originale, Figure 48.a, et l'image marquée figure 51.a est illustrée figure 51.b.



(a) (b)

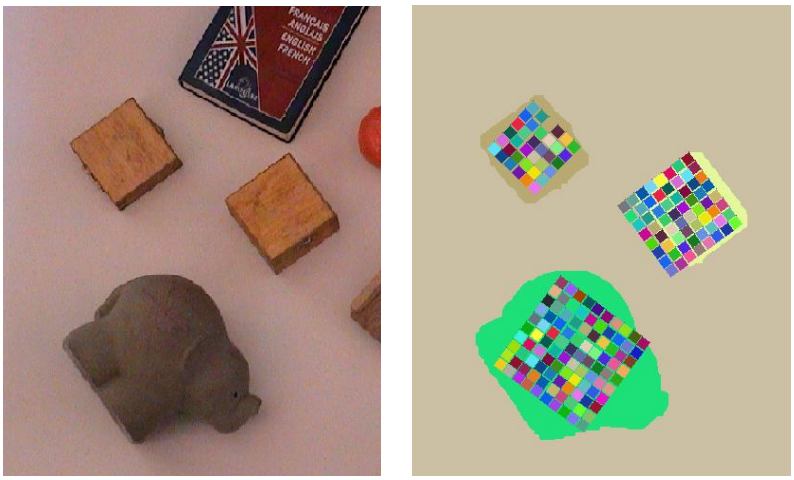
FIG. 51 – *a) Image couleur marquée, b) Différence entre l'image couleur originale et l'image couleur marquée.*

Afin de tester la robustesse de notre méthode à des déformations géométriques, nous avons appliqué à ces deux images des découpages et des rotations. Nous avons également testé notre méthode à la compression JPEG.

Les résultats de la détection du message après un découpage de l'image ont été analysés. La figure 52.a montre le résultat d'un découpage de l'image Objets. Seulement 25 % de l'image est conservée. Dans ce cas, le nombre de RIs diminue et seulement 4 RIs sont détectées. L'ACP est alors appliquée sur cette nouvelle image et la détection des nouveaux parcours de blocs est calculée. Nous obtenons les blocs illustrés figure 52.b. Nous pouvons observer que seulement 3 RIs apparaissent dans cette image. En fait quand une RI est en contact avec les bords de l'image, nous supposons alors que cette RI a été tronquée et que le message extrait sera faux. Par conséquent la détection n'est pas appliquée dans cette RI.

Les résultats de détection après le découpage de l'image sont indiqués dans le tableau 15. Les RIs qui ont disparues, sont bien sûr non détectées et leurs données cachées sont perdues. Dans les RIs présentes quelques erreurs sont détectées. Celles ci sont dues au changement d'espace couleur. Mais après le vote, tous les bits sont correctement détectés.

La figure 51.a montre le résultat d'un découpage sur l'image marquée Fish. Nous obtenons les blocs détectés illustrés sur la figure 51.b. Après le vote, tous les bits sont également correctement détectés. Nous pouvons conclure que la synchronisation effectuée par notre méthode d'IDC résiste au découpage d'image. Notre méthode est donc robuste à ce type de modification géométrique qui appartient à la famille des attaques désynchronisantes.

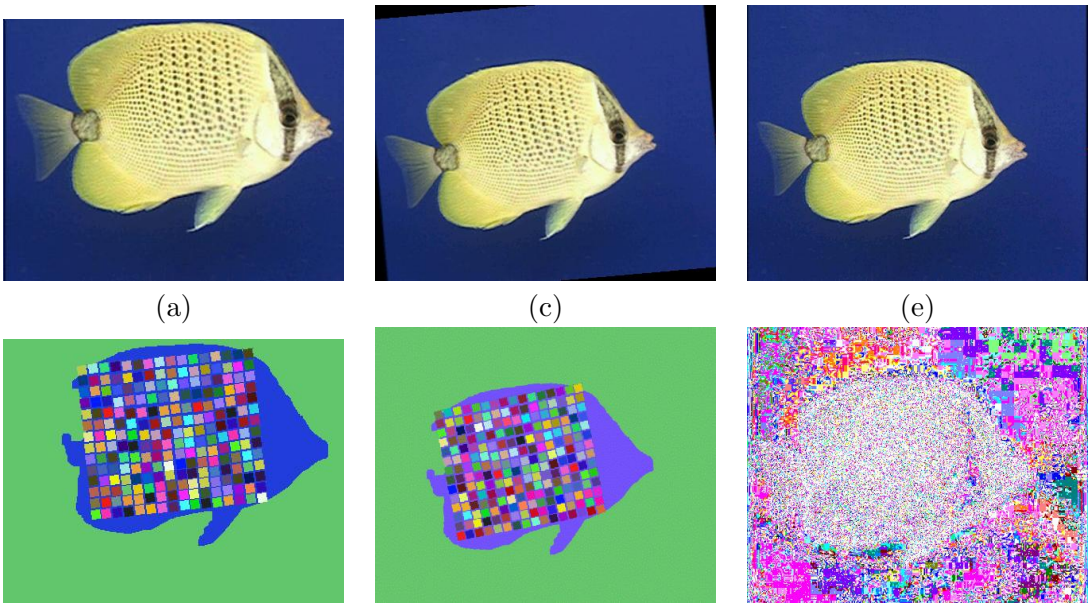


(a) (b)

FIG. 52 – *a) Découpage de l'image marquée, b) Détection et visualisation des blocs marqués.*

RI	% de bits justes (bits justes/bits insérés)			
	Composante Y	Composante Cr	Composante Cb	après vote
1	——	——	——	——
2	——	——	——	——
3	——	——	——	——
4	88,9% (32/36)	94,4% (34/36)	91,7% (33/36)	100% (18/18)
5	——	——	——	——
6	75,0% (48/64)	93,7% (60/64)	93,7% (60/64)	100% (32/32)
7	——	——	——	——
8	76,0% (76/100)	90,0% (90/100)	94,0% (94/100)	100% (50/50)

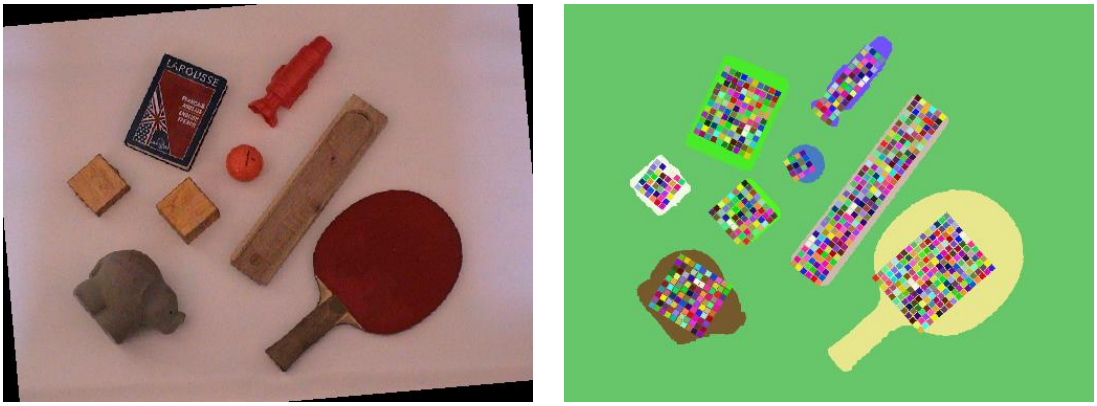
TAB. 15 – *Résultats de la détection des bits après découpage de l'image sur les composantes Y, Cr et Cb marquées et résultats obtenus après le vote.*



(a) (c) (e)

(b) (d) (f)

FIG. 53 – *a) Découpage de l'image marquée, b) Détection et visualisation des blocs marqués, c) Rotation de 5 degrés sur l'image couleur marquée, d) Image étiquette de l'image couleur marquée avec les blocs obtenus après rotation, e) Image couleur marquée comprimée avec JPEG pour un FQ = 80%, f) Différence entre l'image originale et l'image comprimée marquée.*



(a) (b)

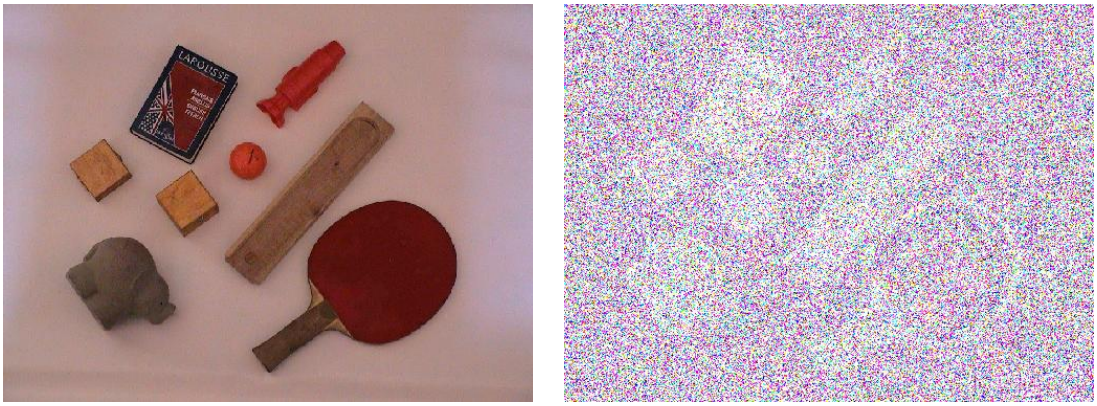
FIG. 54 – *a) Rotation de 5 degrés sur l'image couleur marquée, b) Image étiquette de l'image couleur marquée avec les blocs obtenus après rotation.*

RI	% de bits justes (bits justes/bits insérés)			
	Composante Y	Composante Cr	Composante Cb	après vote
1	92,9% (52/56)	94,6% (53/56)	89,3% (50/56)	100% (28/28)
2	87,5% (105/120)	84,2% (101/120)	94,2% (113/120)	100% (60/60)
3	75,0% (153/204)	86,7% (177/204)	93,6% (191/204)	100% (102/102)
4	87,5% (14/16)	93,7% (15/16)	93,7% (15/16)	100% (8/8)
5	80,6% (29/36)	86,1% (31/36)	88,9% (32/36)	100% (18/18)
6	56,2% (36/64)	93,7% (60/64)	92,2% (59/64)	100% (32/32)
7	81,2% (156/192)	94,3% (181/192)	87,5% (168/192)	100% (96/96)
8	57,0% (57/100)	87,0% (87/100)	94,0% (94/100)	100% (50/50)

TAB. 16 – *Résultats de la détection des bits sur chaque composante Y, Cr, Cb, après une rotation Objets de l'image de 5 degrés.*

Nous allons maintenant évaluer la robustesse de notre méthode par rapport à une autre modification géométrique qui est la rotation. Une rotation de 5 degrés est appliquée sur les deux images marquées figures 50.e et 51.a. Deux images tournées sont obtenues et illustrées figures 54.a et 53.c. Avec cette modification nous devons vérifier la robustesse de la synchronisation entre l'image et les données cachées. Après segmentation, nous obtenons des RIs qui ont tourné et nous commençons l'ACP. Les parcours ainsi que les blocs sont alors construits comme montrés figures 54.b et 51.d. Nous pouvons observer que les axes principaux des RIs sont justes tournées de 5 degrés. La détection des chemins reste donc utilisable. D'un autre coté, si les axes principaux changent, les formes des blocs marqués changent également. Ceci est dû à la structure discrète des images. De plus, la discrétisation crée un autre problème : si l'image est tournée, un ensemble de pixels en bordure de régions vont changer de régions. De ce fait le calcul direct des valeurs des DCT est également modifié. Ce problème de discrétisation est une des raisons qui fait que l'on obtient des erreurs pour les RIs de l'image Objets, tableau 16. Ce tableau donne les pourcentages de bits justes dans chaque RI sur les composantes Y, Cr et Cb. Avec la double redondance, les bits du message sont marqués 6 fois. Un système de vote est alors appliqué et les résultats sont présentés dans la dernière colonne du tableau 16. Dans chaque RI, le message caché est correctement détecté. Par conséquent, nous pouvons en conclure que notre synchronisation résiste aux rotations.

Nous appliquons maintenant l'algorithme JPEG couleur sur les images marquées afin de tester la robustesse à la compression. La figure 55.a montre l'image marquée comprimée avec un facteur de qualité égal à 80 %. Pour diminuer la taille de l'image, l'algorithme



(a) (b)

FIG. 55 – *a) Image marquée et comprimée avec FQ = 80%, b) Différence entre l'image originale et l'image marque et comprimée avec FQ = 80%.*

RI	% de bits justes (bits justes/bits insérés)			
	Composante Y	Composante Cr	Composante Cb	après vote
1	69,6% (39/56)	87,5% (49/56)	75,0% (42/56)	100% (28/28)
2	75,8% (91/120)	87,5% (105/120)	81,7% (98/120)	100% (60/60)
3	75,5% (154/204)	88,2% (180/204)	88,7% (181/204)	100% (102/102)
4	72,2% (26/36)	94,4% (34/36)	83,3% (30/36)	100% (18/18)
5	68,8% (11/16)	75,0% (12/16)	87,5% (14/16)	100% (8/8)
6	50,0% (32/64)	81,3% (52/64)	82,8% (53/64)	100% (32/32)
7	75,5% (145/192)	76,0% (146/192)	81,25% (156/192)	100% (96/96)
8	50,0% (50/100)	82,0% (82/100)	88,0% (88/100)	100% (50/50)

TAB. 17 – *Résultats de la détection des bits après compression sur chaque composante Y, Cr, Cb.*

JPEG modifie les couleurs des pixels. Avec notre méthode, la détection est correcte si le FQ est supérieur à 75 %. En dessous de cette valeur, le bruit dû à la compression devient trop important. En effet, l'IDC et la compression ne sont plus effectués en même temps car les blocs de la compression et de l'IDC sont différents (forme et taille). Le tableau 17 donne le pourcentage de bits justes détectés dans chaque RI sur les trois composantes. Nous observons que les messages sont correctement détectés grâce à la redondance et au système de vote.

La figure 55.b montre la différence entre l'image comprimée et marquée figure 55.a et l'image originale figure 48.a. Avec cette image différence, l'invisibilité de l'IDC est illustrée, le PSNR est égal à 43,19 dB.

La figure 53.e montre l'image comprimée marquée Fish avec un FQ= 80 %. La figure 53.f montre la différence entre l'image comprimée marquée figure 53.e et l'image originale figure 51.a. Le message est correctement détecté par l'intermédiaire du système de vote. Notre méthode d'IDC résiste donc à la compression JPEG pour des bons FQ.

Dans cette section nous avons présenté une méthode d'IDC couleur qui utilise le contenu des images. Afin d'obtenir la synchronisation entre le message caché et l'image, une analyse est faite et plusieurs RIs sont crées. Le contenu de l'image est utilisé pour synchroniser le message et l'image. Les trois composantes couleur Y, Cr et Cb sont utilisées pour insérer trois fois le message. Ceci est le premier degré de redondance. De plus, chaque bit est dupliqué et embarqué deux fois dans chaque RI. Ceci correspond au second niveau de redondance. Au niveau résultat, nous avons montré la robustesse de notre méthode par rapport à des transformations géométriques et que malgré tout la méthode reste robuste à la compression JPEG.

Les différents résultats illustrent le fait que la méthode d'IDC dépend de la segmentation couleur de l'image. Notre objectif principal a pu être atteint par l'intermédiaire de cette nouvelle méthode. En effet, notre méthode est robuste à un ensemble de traitements tels que rotation, découpage et compression. Les données embarquées restent invisibles, de plus le bruit apporté par la méthode d'IDC reste négligeable par rapport aux autres modifications telle que la compression.

Tous les résultats ont été obtenus avec des blocs de taille environ $\mathbf{11 \times 11}$ pixels. Si la taille des blocs est plus petite, le nombre de bits insérés augmente mais la robustesse diminue. Au contraire si la taille des blocs augmente alors la robustesse est améliorée mais la taille des données cachées devient plus petite. Pour améliorer la quantité des données cachées, nous pensons adapter la taille des blocs marqués à la forme des RIs. Tous les blocs ne seront pas forcément carrés et le taux d'insertion augmentera. Comme perspectives, nous envisageons également de changer la taille des blocs par rapport à la taille de chacune des RIs afin que la méthode devienne robuste au zoom.

3.7 Conclusion et perspectives

Dans ce chapitre nous avons développé des nouveaux algorithmes d'IDC robustes à la compression permettant de protéger des données durant le transfert des images sur les réseaux. Dans un premier temps, après avoir présenté des méthodes existantes d'IDC et en particulier celles basées sur la DCT, nous avons présenté une nouvelle méthode

d'IDC par bloc avant quantification. Cette méthode permet de résister à la compression et permet d'insérer une quantité importante de données cachées. Nous avons appliqué notre amélioration aux méthodes existantes et avons montré que dans tous les cas la qualité était améliorée. Finalement, dans ce chapitre, nous avons étendu notre méthode d'IDC aux RIs contenues dans les images afin de rendre notre méthode robuste aux rotations et aux découpages. Nous avons également utilisé des images couleurs pour rajouter de la redondance et donc pour être plus robuste à l'ensemble des transformations que peut subir une image. Nous sommes conscient que le challenge d'insérer une quantité d'information relativement importante dans une image et d'être également robuste à des transformations synchrones et asynchrones est difficile. Nous travaillons actuellement dans cette direction en utilisant en particulier les caractéristiques discrètes des images.

Chapitre 4

Cryptage d'images

Dans ce chapitre, nous montrons comment les algorithmes classiques de chiffrement peuvent être appliqués à des images. Les données images sont des données particulières du fait de la taille des images et de l'information bidimensionnelle. Nous présentons de nombreux algorithmes par bloc ou par flot symétrique ou asymétrique. Nous concluons que les algorithmes asymétriques tel que le RSA ne sont pas adaptés aux images du fait de leur complexité dû à l'utilisation de grands nombres premiers car une partie de la clef est connue (clef publique). Concernant les algorithmes symétriques, les méthodes par bloc présentent des inconvénients quand l'image contient des zones homogènes. Dans le cas des algorithmes de chiffrement par flot, les zones homogènes ne sont plus visibles dans l'image cryptée. De plus les chiffrements par flot sont très rapides. Cependant, quelque soit l'algorithme de cryptage utilisé, il est alors difficile de comprimer l'image puisque théoriquement les redondances ont été supprimées durant la phase de cryptage et donc l'entropie devient maximale. De plus les algorithmes de chiffrement par bloc supportent très mal le bruit, en effet dès qu'un bit d'un bloc est altéré alors le bloc complet n'est pas décryptable. Dans le cas des chiffrements par flot, la robustesse au bruit semble plus importante. Dans ce chapitre nous présentons également une première approche de crypto-compression basée sur des images contenant des zones homogènes. Le premier objectif de cette méthode était de faire disparaître les zones homogènes, mais au final l'image est comprimée sans perte.
Les analyses développées dans ce chapitre sont à la base des méthodes de codage conjoint que nous présentons dans le chapitre suivant. L'objectif est alors de combiner les processus de compression et de cryptage.

Ces travaux ont été développés avec **S. Piat** et **G. Benoît** dans le cadre de leur stage de DEA ainsi qu'avec **JC. Borie** dans le cadre de sa thèse.

Cette partie a donné lieu aux publications suivantes : [Puech 01c, Puech 01b, Puech 01a, Puech 01d, Borie 02a, Borie 02b, Borie 04b, Borie 04a].

4.1 Introduction

Nous présentons dans ce chapitre des algorithmes de cryptage appliqués à la sécurisation des images. Aujourd'hui, par exemple, lorsqu'un médecin reçoit un patient il a souvent besoin de l'avis d'un spécialiste avant de prononcer son diagnostic. La solution serait de transmettre les images qu'il a obtenues par liaison informatique. Mais les réseaux informatiques sont complexes et les écoutes illégales nombreuses. Il se pose donc un réel problème quant à la sécurité lors de la transmission de données. Un domaine aussi important et secret que la médecine par exemple ne peut donc se permettre de prendre un tel risque sans se protéger. La protection la plus adaptée pour ce type de communication résiderait dans la cryptographie. Beaucoup de techniques de cryptage de texte ont été développées.

Depuis l'antiquité, les hommes ont toujours essayé de coder des messages secrets pour se prévenir des oreilles malveillantes. César, pour communiquer ses ordres et rapports à ses centurions, avait eu l'idée de remplacer chaque lettre de ses messages par la troisième lettre suivante dans l'alphabet [Schneier 97]. Ainsi le A devenait D, B devenait E, ... Plus tard, l'abbé Trithème, les détenus nihilistes russes, Della Porta, Vigenère, Cardano, les Allemands avec Enigma et bien d'autres apportèrent leurs idées en créant des codes secrets permettant de chiffrer des messages. C'est ainsi qu'est née la cryptographie.

Dans les premières esquisses de cette science du secret, la sécurité résidait dans la confidentialité de l'algorithme qui permettait le chiffrement et le déchiffrement. C'est au fil du temps qu'est apparue progressivement la notion de clef. Aujourd'hui, les systèmes de cryptage reposent sur des algorithmes mis à disposition de tous et c'est la clef, code secret particulier, qui est confidentielle et qui permet de crypter ou de décrypter le message [Kerckhoffs 83].

Les processus de cryptage utilisent soit une clef identique pour le chiffrement et le déchiffrement, soit des clefs différentes [Diffie 76, Stinson 95]. Ces processus sont symétriques ou asymétriques et sont appliqués par bloc ou par flot de données. De nombreux algorithmes utilisent des grands nombres premiers afin de garantir la confidentialité du fait qu'une partie de la clef (clef publique) est connue. Nous allons voir que ce type d'algorithme n'est pas approprié pour une application à des images car ils nécessitent un long temps de

calcul. Il existe quatre grands objectifs pour le cryptage, utilisés à des fins différentes :

- Tout d'abord, **la confidentialité** ou masquage des données, caractéristique la plus utilisée, vise à rendre le cryptogramme inintelligible pour celui qui n'est pas en possession de la clef.
- Ensuite, **l'authentification** permet à l'émetteur de signer son message. Ainsi, le récepteur n'aura pas de doute sur l'identité du premier.
- **L'intégrité** quant à elle va assurer au récepteur que le contenu du message n'a pas pu être malencontreusement falsifié depuis son écriture.
- Enfin, **la non-répudiation** est la garantie qu'aucun des deux individus ayant effectué une transaction ne pourra nier avoir reçu ou envoyé les messages.

La caractéristique essentielle qui nous intéresse dans cette section est essentiellement la première, à savoir la confidentialité (nous reviendrons sur la caractéristique intégrité ainsi que sur les deux autres dans le chapitre 5).

Les algorithmes de cryptage peuvent donc être séparés en fonction de plusieurs caractéristiques : les systèmes à clefs publique-privée (systèmes asymétriques) et ceux à clef secrète (systèmes symétriques) [Diffie 76, Stinson 96]. Les systèmes à clef secrète sont ceux qui permettent de crypter et de décrypter avec la même clef. Il va de soi que l'émetteur et le récepteur doivent s'être auparavant partagé le secret de la clef par un moyen de communication sécurisé.

Les systèmes à clef publique ou asymétriques permettent de pallier à cette incommodité en utilisant une clef pour crypter, et une autre clef pour décrypter. Chaque individu X détiendra un couple de clefs, dont une sera confidentielle (la clef privée) et l'autre connue de tous (la clef publique). Pour écrire à X, il suffit de chiffrer le message avec la clef publique de X que l'on connaît. A la réception, seul X pourra déchiffrer avec sa clef privée.

De nos jours, de plus en plus d'images numériques sont transférées sur les réseaux informatiques. Aucun système particulier n'a été développé pour le transfert sécurisé d'images. Dans cette section, nous proposons donc d'appliquer à des images des algorithmes classiques de chiffrement afin d'effectuer des transferts sécurisés d'images. Nos travaux reposent sur l'idée d'adapter ces algorithmes à l'information image qui reste un signal particulier de par sa redondance et de par sa nature bidimensionnelle. Nous proposons de chiffrer des images au niveau des codages source afin de faire remonter cette fonctionnalité au niveau des couches hautes. De cette manière, la fonctionnalité cryptage d'images peut être insérée au niveau d'un logiciel. Le problème consiste ensuite à faire résister ces

chiffrements à des traitements avals comme la compression. En effet, la quantité d'information (entropie) à transmettre augmente fortement entre l'image originale et l'image cryptée. Dans le cas particulier de certains types d'images médicales, des grandes zones homogènes apparaissent. Ces zones perturbent l'efficacité des algorithmes de chiffrement. Nous analyserons alors ce problème en associant une compression de l'image initiale au chiffrement.

Dans la section 4.2 nous décrivons les algorithmes classiques de cryptage. Nous montrons, section 4.3, comment adapter ces algorithmes aux données images. Dans la section 4.4 nous commentons les différents résultats obtenus sur des images en fonction des algorithmes de cryptage utilisés. Nous comparons, section 4.5, le comportement à la compression des algorithmes de cryptage des images. Nous expliquons section 4.6, comment il est possible, dans le cas particulier d'images contenant des grandes zones homogènes, de réaliser en même temps une compression et un cryptage d'images.

4.2 Algorithmes de chiffrement

Nous avons étudié plusieurs systèmes de cryptage de données pour les appliquer aux images. Afin d'explorer une zone assez étendue de ce qui se fait en cryptage, nous avons choisi quatre systèmes assez hétérogènes par bloc à clef secrète (DES et TEA)[1,2], par bloc à clef publique (RSA) et un système de chiffrement par flot [Piat 03].

4.2.1 Système par bloc symétrique : DES

Nous nous sommes orientés dans le cadre du cryptage symétrique vers deux algorithmes de chiffrement existants à savoir DES et TEA. Ils font tous deux partie de la catégorie des systèmes de chiffrement par blocs. Dans cette section nous présentons l'algorithme DES (Data Encryption Standard).

En 1973, le NIST (National Institute of Standards and Technology) fait un appel d'offres public pour un algorithme de cryptage devant répondre à plusieurs critères. Il devait avant tout être destiné à devenir un standard utilisé par tous, et sa sécurité résiderait uniquement dans le secret de la clef, l'algorithme devant être public. C'est en 1974, après un second appel d'offres, que IBM propose un algorithme du nom de LUCIFER. Cet algorithme

1. Au moment où j'ai réalisé ces travaux, l'algorithme AES n'avait pas encore pris le dessus sur le DES. Dans le chapitre suivant, chapitre relatif à des travaux plus récents, seul l'AES sera comparé à nos propres algorithmes.

2. Notons cependant que l'AES en mode par bloc (ECB Mode) s'applique absolument de la même manière sur les images que l'algorithme DES présenté dans ce chapitre.

deviendra peu de temps après le DES, premier standard de la cryptographie moderne [Schneier 97].

L'algorithme DES repose sur 16 rondes au cours desquelles un bloc de données de 64 bits va se mélanger avec la clef K, qui est codée sur 64 bits également. A chacune de ces rondes on crée une sous-clef k_i qui est calculée à partir de la clef de départ K, et c'est avec cette sous-clef que l'on va brouiller le bloc [NBS 77].

4.2.1.1 Calcul des sous-clefs

En ce qui concerne le calcul des sous-clefs, on commence d'abord par appliquer à K une permutation de 56 de ses 64 bits. Cette opération a pour effet de mélanger les bits de la clef et d'amener celle-ci à une taille réduite de 56 bits.

Puis à chacune des 16 rondes, on sépare la clef en deux blocs de 28 bits. Ces derniers subiront des rotations de bits dans un sens et dans l'autre, avant d'être réunis et repermutés, pour enfin donner la sous-clef k_i (figure 56).

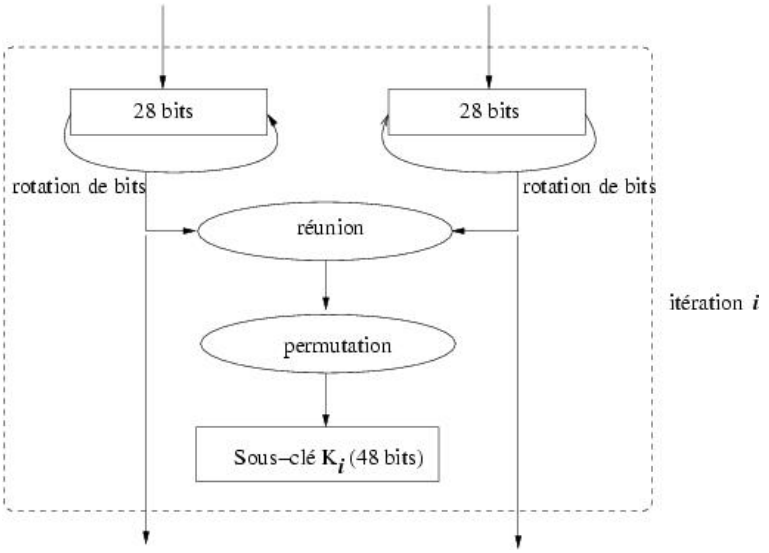


FIG. 56 – *Calcul des sous-clefs du DES.*

4.2.1.2 Cryptage/décryptage du bloc

Une fois les 16 sous-clefs générées à partir de notre clef secrète, nous allons chiffrer (ou déchiffrer) un bloc de données de 64 bits. Le processus commence par une permutation initiale (IP) qui modifie l'ordre des bits du bloc de départ, avant de scinder le résultat en deux blocs de 32 bits, L_0 et R_0. De même que pour le calcul des sous-clefs, on effectue alors

16 rondes, où les deux sous-blocs L_i et R_i vont être modifiés en fonction de la sous-clef K_i.

A chaque itération i, L_i prend la valeur de R_{i-1}, alors que R_i devient $L_{i-1} \oplus P(S_i(E(R_{i-1}) \oplus K_i))$ [Guillem 02]. L'opérateur $\oplus$ est le *ou exclusif*, P est une permutation, S_i une substitution par *S-box* et E l'opération d'expansion (figure 57).

E est une modification proche d'une permutation qui va faire passer notre bloc R_{i-1} de 32 à 48 bits.

Interviennent alors les boîtes de substitution (S-Boxes). Nous disposons d'un bloc de 48 bits. Découpons-le en 8 mots (de B_1 à B_8) de 6 bits $b_1b_2b_3b_4b_5b_6$. Chaque mot B_j va subir une transformation par la S-box numéro j. Une S-box est un tableau à deux dimensions (4 lignes et 16 colonnes) contenant des nombres compris entre 0 et 15. Le mot b_1b_6 nous donne la ligne de la S-box à prendre en compte, et $b_2b_3b_4b_5$ la colonne. On récupère alors le nombre contenu dans la case correspondante. Compris entre 0 et 15, on le code sur 4 bits. Les 8 résultats des S-boxes appliquées aux B_j concaténés, on obtient un nouveau bloc de 32 bits. A la fin de chaque itération on fait subir à ce bloc de 32 bits une permutation P.

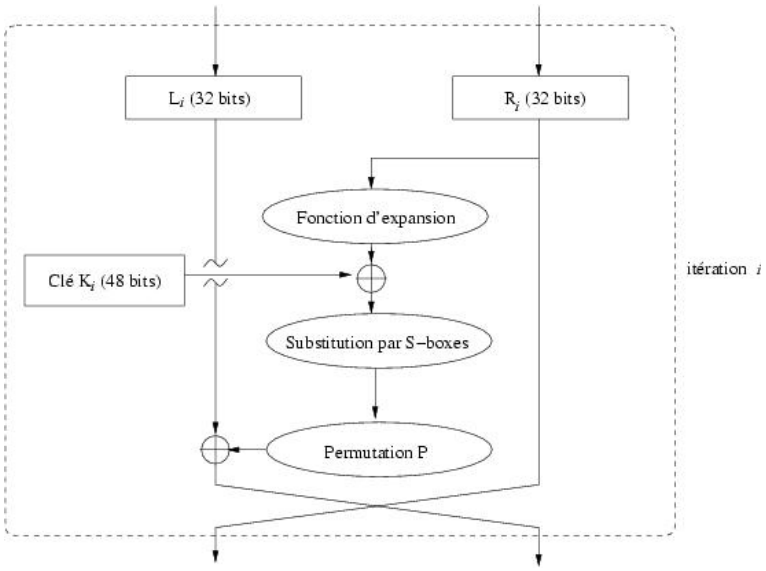


FIG. 57 – *Chiffrement d'un bloc avec DES.*

Enfin, les 16 rondes passées, avant de retourner le résultat on fait passer une dernière fois le bloc par une permutation finale (FP). Cette permutation n'est autre que l'opération inverse de la permutation initiale IP.

Pour le décryptage, le procédé est le même, mis à part que les sous-clefs sont utilisées

dans l'ordre inverse (de la seizième à la première).

Aujourd'hui, un peu plus de vingt ans plus tard, même si l'algorithme est toujours aussi robuste il souffre quelque peu du fait de la longueur de sa clef limitée à 64 bits. En effet, la performance actuelle des machines en termes de rapidité de calcul rend le DES cassable. L'attaque, dite brutale, qui consiste à essayer toutes les 2^{64} clefs potentielles est désormais abordable par de gros calculateurs. Une solution a été apportée pour augmenter la sécurité ; elle s'appelle le triple-DES. Le triple-DES consiste à crypter le bloc d'entrée 3 fois avec des clefs différentes K_1, K_2 et K_3. Il existe plusieurs variantes, mais en général les première et troisième opérations sont des opérations de cryptage, tandis que la seconde est une opération de décryptage (figure 58). Souvent, on choisit également $K_1 = K_3$, ce qui permet à la clef totale de ne pas dépasser 128 bits.

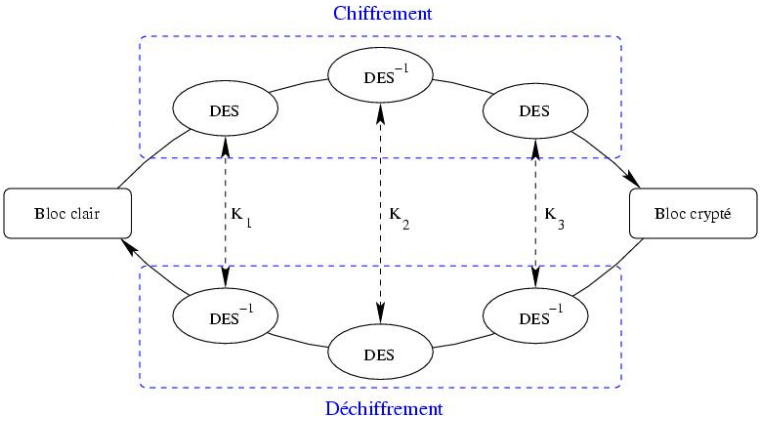


FIG. 58 – *Principe de chiffrement/déchiffrement avec le triple-DES.*

Depuis 4 ans maintenant le DES est remplacé progressivement par l'AES. Le principe de l'AES est très proche de celui du DES mais en plus de sa modernité, il propose des longueurs de clefs comprises entre 128 bits et 192 bits.

4.2.2 Système par bloc symétrique : TEA

Pour le TEA (Tiny Encryption Algorithm), il s'agit là encore d'un algorithme de chiffrement à clef secrète, où la taille des blocs est de 64 bits, mais où la clef est codée sur 128 bits. Bien que beaucoup moins célèbre que le DES, aucune attaque sérieuse ne lui a révélé de faille à ce jour.

Le principe de TEA réside dans un petit nombre d'opérations mêlant clef et données, répété en boucle sur un nombre n d'itérations [Wheeler 94]. Bien qu'une valeur de n égale

à 8 soit jugée suffisamment sûre, il est recommandé d'utiliser plutôt $n = 32$ par exemple, l'algorithme TEA étant particulièrement rapide.

Soit δ le nombre d'or ($\frac{\sqrt{5}+1}{2}$), auquel on retranche 1 pour obtenir un nombre compris entre 0 et 1. On multiplie le tout par 2^{32} et on récupére que la partie entière. Cela nous donne un nombre entier sur 32 bits dont la répartition de 1 et de 0 est assez arbitraire.

Voici maintenant le calcul qui est effectué à chaque itération i (allant de 1 à n). On utilise un entier qui est le produit de i par δ, et on divise la clef et le bloc respectivement en quatre (a,b,c et d) et en deux (y et z) nombres de 32 bits.

On réalise alors sur y et z des opérations basées sur le ou exclusif ($\oplus$), la somme et le décalage de bits (à l'aide des opérateurs << et >>). Ces opérations sont :

$$y_{i+1} \leftarrow y_i + ((z_i << 4) + a) \oplus (z_i + \delta \times i) \oplus ((z_i >> 5) + b)$$
$$z_{i+1} \leftarrow z_i + ((y_{i+1} << 4) + c) \oplus (y_{i+1} + \delta \times i) \oplus ((y_{i+1} >> 5) + d)$$

Au terme des n itérations, le regroupement de y et z formera notre bloc de 64 bits crypté. La figure 59 illustre l'ensemble des opérations effectuées dans chacune des rondes. Pour ce qui est du décryptage, il suffit de reprendre l'algorithme à contre-courant, en

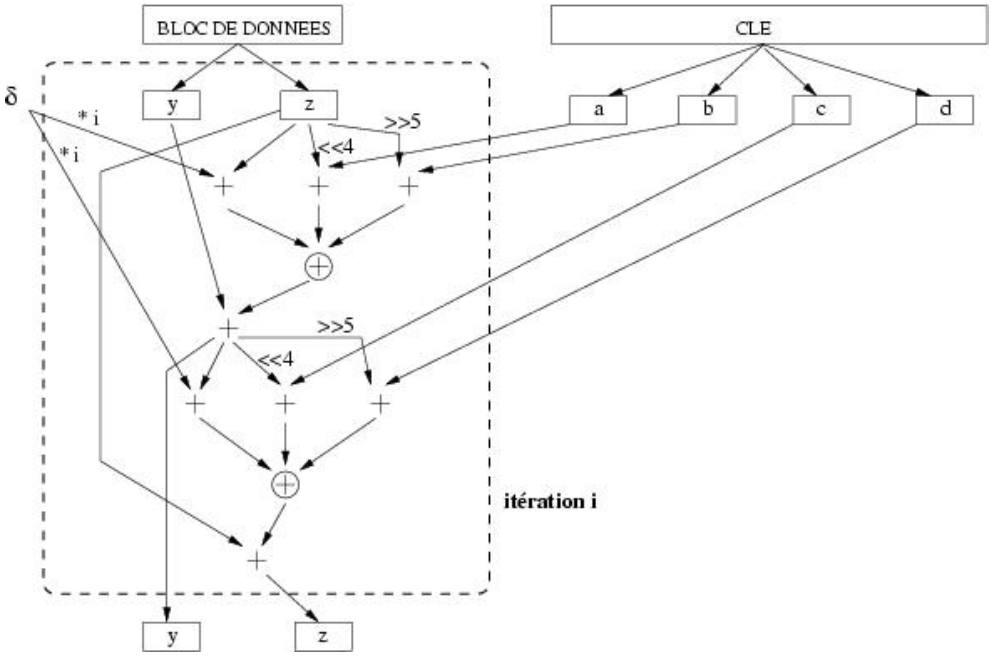


FIG. 59 – *Chiffrement d'un bloc avec TEA.*

faisant aller les numéros de rondes i de n à 1, on effectue d'abord l'affectation en z puis celle en y et on leur retranche les valeurs que nous y avons ajoutées lors du cryptage. Nous

avons donc là un système de cryptage à clef secrète dont la sécurité a certes été moins étudiée que celle de DES, mais qui a l'avantage d'offrir une clef de 128 bits et d'être très rapide, compte tenu du faible nombre d'opérations.

4.2.3 Système par bloc asymétrique : RSA

Nous nous intéressons maintenant aux systèmes à clef publique. L'algorithme RSA est le plus célèbre de sa catégorie. Sa sécurité réside dans le fait qu'il est matériellement trop lent de factoriser des très grands nombres en produits de facteurs premiers [Schneier 95, Rivest 78].

Soient p et q deux très grands nombres premiers distincts, et n un très grand nombre qui est le produit de p et q. On note $\phi(n)$ la fonction d'Euler en n, qui est le nombre de nombres entiers naturels inférieurs à n et premiers avec n. Si l'on note N l'ensemble de ces nombres, on cherche à calculer le cardinal de N, noté $card(N)$. Notons également P et Q les ensembles respectifs des multiples de p et de q compris dans [1,$n-1$]. p et q étant les deux facteurs de n, on peut affirmer que $card(P) = q - 1$, et que $card(Q) = p - 1$. De plus, p et q sont premiers, donc P et Q sont deux ensembles disjoints (et inclus dans [1,$n-1$]. Donc finalement,

$$\begin{aligned}
\phi(n) &= card([1,n-1]\backslash P\backslash Q) \\
&= (n-1) - card(P) - card(Q) \\
&= (pq-1) - (q-1) - (p-1) \\
&= pq - q - p + 1 \\
&= (p-1)(q-1).
\end{aligned}$$

La paire clef publique / clef privée va résider dans deux nombres d et e associés à n. e est d'abord calculé aléatoirement entre 2 et $\phi(n)$ et doit être premier avec $\phi(n)$. C'est la paire (n,e) qui constitue la clef publique. Puis d est calculé tel que $d = e^{-1} mod(n)$. L'algorithme d'Euclide étendu permet de calculer cet inverse instantanément, même avec de très grands nombres. (n,d) constitue la clef privée.

L'usage des clefs dans le cryptage et le décryptage est le suivant. Si m est le message clair (inférieur à n, sinon on le découpe), on le crypte avec la clef privée (n,e) en l'élevant à la puissance e, modulo n. On obtient le message chiffré $m' = m^e mod(n)$.

Pour le décryptage, nous avons besoin de la seconde clef (n,d). En élevant le message crypté à la puissance d modulo n, on obtient :

$$(m')^d mod(n) = (m^e mod(n))^d mod(n) = m^{ed} mod(n) = m$$

car d et e sont inverses modulo n.

Par exemple, si Bob souhaite envoyer un message à Alice, il convertit son message en nombre et coupe le message en blocs de taille plus petite que n. Pour chaque bloc m_i, en utilisant la clef publique d'Alice, Bob calcule et chiffre le bloc de la manière suivante :

$$c_\imath = m_\imath^e mod\ n, \tag{64}$$

avec $\imath$, la position du bloc dans le texte, $\imath \in [1,N]$, si N est le nombre de blocs.

Alice, avec sa clef privée, peut alors décrypter le message en faisant :

$$m_\imath = c_\imath^d\ mod\ n. \tag{65}$$

Ainsi, la méthode RSA se distingue des systèmes de chiffrement symétriques de par l'utilisation de deux clefs différentes pour le cryptage et le décryptage. L'une de ces deux clefs, la clef publique, sera censée être connue de tous, et l'autre, la clef privée, connue par un seul individu. L'algorithme RSA peut permettre soit de crypter avec une clef publique, dans lequel cas seul le destinataire pourra déchiffrer le message avec sa clef privée, soit de crypter avec sa propre clef privée. Dans ce cas, tout le monde peut lire le message grâce à la clef publique, mais l'émetteur a pu authentifier le message puisqu'il est potentiellement le seul à avoir pu crypter avec sa clef privée. Souvent, on effectue un double cryptage clef publique - clef privée pour combiner les avantages des deux méthodes. Une sécurité supplémentaire peut être ajoutée si Bob chiffre une première fois le message avec sa clef privée puis une seconde fois avec le clef publique d'Alice. A la réception, après avoir utilisé sa clef privée, Alice devra alors utiliser la clef publique de Bob.

Malheureusement, RSA est un algorithme très lent, beaucoup plus lent que n'importe quel système symétrique, et d'autant plus que les nombres utilisés sont grands. De plus, il est aujourd'hui facilement cassable, même pour des nombres de 512 bits. Sa lenteur fait qu'il est préférable de l'utiliser pour envoyer de manière sécurisée une clef secrète, qui permettra de déchiffrer le message, avec DES par exemple, plus rapide que RSA.

4.2.4 Algorithmes de chiffrement par flot

Les algorithmes de chiffrement par flot peuvent être définis comme étant des algorithmes de chiffrement par bloc, où chaque bloc est de dimension unitaire (1 bit, 1 octet, ...) ou relativement petit. Leurs principaux avantages sont leur extrême rapidité et leur capacité à changer à chaque symbole du texte clair.

Avec un algorithme de chiffrement par flot, il est possible de crypter séparément chaque caractère du message clair un par un, en utilisant une transformation du cryptage qui

varie à chaque fois. De plus, la fonction de cryptage peut varier au fur et à mesure que le message clair est traité. Par conséquent, les algorithmes de chiffrement par flot ont besoin de mémoires. Généralement, les algorithmes de chiffrement par flot sont composés de deux étapes : la génération d'une clef dynamique et la fonction de cryptage de sortie dépendant de la clef dynamique.

Quand la clef dynamique est générée indépendamment du texte clair et du texte chiffré, l'algorithme de chiffrement par flot est dit synchrone. Avec un chiffrement par flot, l'émetteur et le récepteur doivent se synchroniser en utilisant la même clef et en l'utilisant à la même position. Les chiffrements par flot synchrone sont utilisés fréquemment dans des environnements où les erreurs sont fréquentes car ils ont l'avantage de ne pas propager le bruit [Guillem 02]. Concernant les attaques actives comme l'insertion, la suppression et la copie de digits du texte chiffré par un adversaire actif, celles-ci produisent immédiatement une perte de synchronisation. Le processus de cryptage d'un chiffrement par flot synchrone est décrit figure 60.a, où $f()$ est la fonction qui détermine l'état suivant, $g()$ est la fonction génératrice de la clef dynamique et $h()$ la fonction de sortie de cryptage :

$$\begin{cases} s_{i+1} &= f(K,s_i), \\ z_i &= g(K,s_i) \\ c_i &= h(z_i,m_i), \end{cases} \tag{66}$$

où K est la clef, s_i, m_i, c_i et z_i sont respectivement le i^{eme} état, le texte clair, le texte chiffré et la clef dynamique. Le processus de décryptage est illustré figure 60.b.

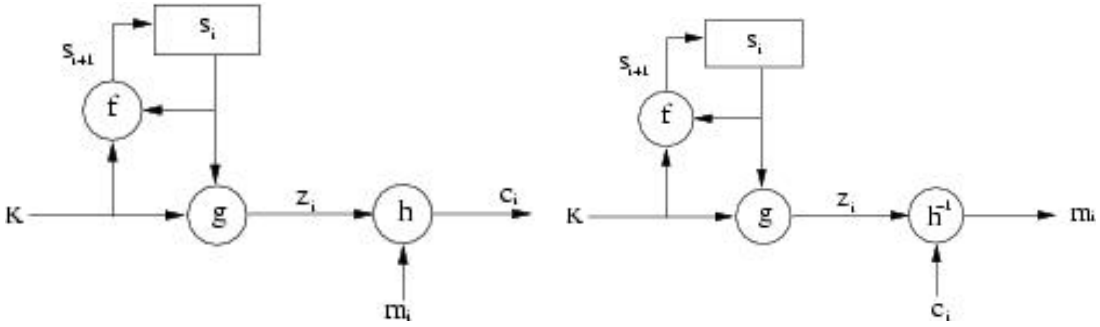


FIG. 60 – *Chiffrement par flot synchrone a) Cryptage, b) Décryptage.*

Quand la clef dynamique est générée à partir de la clef et d'un certain nombre de digits précédemment crypté, l'algorithme de chiffrement par flot est dit asynchrone, appelé aussi chiffrement par flot auto-synchronisant. La propagation des erreurs est limitée à la taille de la mémoire. Si des digits du texte chiffré sont effacés ou insérés en plus, le récepteur

est capable avec la mémoire de se resynchroniser avec l'émetteur. Concernant les attaques actives, si un adversaire actif modifie une part des digits du texte chiffré, le récepteur est capable de la détecter. Le processus de cryptage d'un chiffrement par flot asynchrone est décrit figure 61.a, où $g()$ est la fonction génératrice de la clef dynamique et $h()$ la fonction de sortie de cryptage :

$$\begin{cases} z_i &= g(K,c_{i-t},c_{i-t+1},\ldots,c_{i-2},c_{i-1}) \\ c_i &= h(z_i,m_i), \end{cases} \tag{67}$$

où K est la clef, m_i, c_i et z_i sont respectivement le i^{eme} texte clair, le texte chiffré et la clef dynamique.

Nous pouvons remarquer équations (67) que la clef dynamique dépend des t digits précédents du texte chiffré. Afin d'être robuste à de nombreuses attaques statistiques, la fonction génératrice de la clef dynamique $g()$ doit produire une séquence d'une large période avec de bonnes propriétés statistiques qui peuvent être appelées séquences binaires pseudo-aléatoires (SBPS). Le processus de décryptage est illustré figure 61.b.

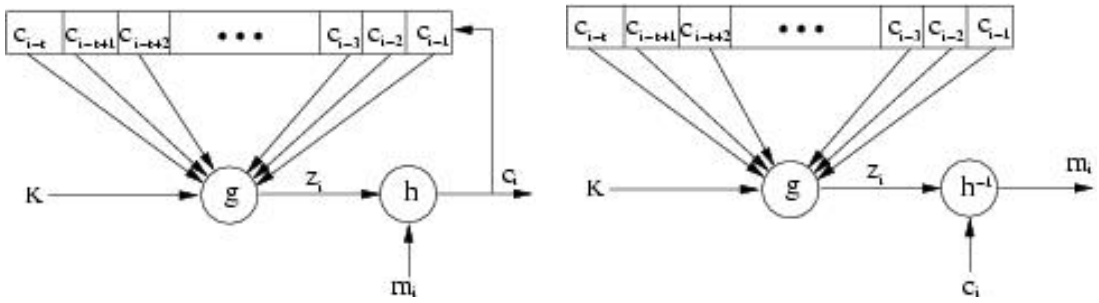


FIG. 61 – *Chiffrement par flot asynchrone a) Cryptage, b) Décryptage.*

Le Linear Feedback Shift Register (LFSR), illustré figure 62, est un mécanisme très souvent utilisé dans les chiffrements symétriques de flux. Il génère des séquences de bits pseudo-aléatoires. La série de bits, appelée registre, est initialisée par un vecteur d'initialisation qui est la plupart du temps la clef du chiffrement. Le comportement du vecteur "registre" est défini par rapport à un compteur : à chaque itération de la boucle, le contenu du registre est décalé vers la droite d'une position et l'opération du *ou exclusif* est appliquée sur un sous-ensemble de bits (choisi selon l'algorithme), dont le résultat est placé à l'extrême gauche du registre. A la fin de chaque itération, un bit de sortie est généralement gardé pour former le régistre transformé résultant.

Les LFSR sont très rapides et faciles d'implémentation autant dans des logiciels que sur du matériel. Les algorithmes de chiffrement de flux sont beaucoup moins nombreux que

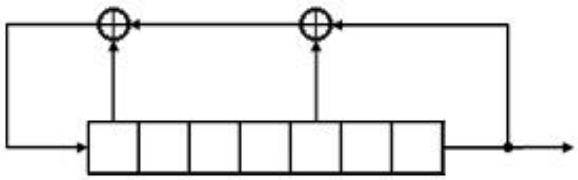


FIG. 62 – *Mécanisme très souvent utilisé dans les chiffrements symétriques par flot (Linear Feedback Shift Register (LFSR).*

ceux de chiffrement par blocs. Cependant, leur popularité est croissante avec la quantité toujours grandissante d'informations circulant sur les réseaux comme Internet et étant traitée par les logiciels. C'est dans ce domaine des logiciels que les chiffrements par flot ont toute leur importance. Actuellement, les deux principaux sont les algorithmes RC4 et SEAL. L'algorithme RC4 a été pensé par Ron Rivest en 1987 et développé pour la RSA Security. Il est basé sur les permutations aléatoires, avec des opérations sur des octets. L'algorithme a une longueur de clef variable (de 1 à 256 octets). La clef est utilisée pour initialiser une "table d'états" de 256 octets. La table d'états est employée pour la génération d'octets pseudo-aléatoires puis pour produire le flux pseudo-aléatoire avec lequel le texte clair sera transformé en utilisant l'opération du *ou exclusif*. Le RC4 est employé dans plusieurs applications commerciales, par exemple dans le protocole SSL et dans Oracle Secure SQL car l s'exécute très rapidement.

4.3 Application des algorithmes de chiffrement aux images

Dans cette section, nous montrons comment il est possible d'appliquer les algorithmes présentés précédemment à des images en niveaux de gris. Nous considérons donc une image composée de N pixels d'un octet. Notre objectif de chiffrement d'images est d'obtenir une image de même format et de taille égale au maximum à la taille de l'image originale. Le chiffrement doit se passer avec le minimum de perte, voir même sans aucune perte pour certaines applications telle que l'imagerie médicale. Nous traitons donc le chiffrement d'images comme un codage source de manière à traiter cette fonctionnalité au niveau de l'application. De ce fait, si un utilisateur n'a pas la bonne clef, il a accès au moins à une image dans un format connu. En remontant le chiffrement au niveau de l'application, il est alors possible de procéder, par exemple, à un fenêtrage d'image. Dans le cas d'images de grande taille, il n'est alors pas utile de décrypter toute l'image si l'on ne souhaite en visualiser qu'une zone particulière.

4.3.1 Chiffrement d'images par DES ou par TEA

Dans le cas de ces deux algorithmes par bloc la longueur des blocs est imposée et est égale à 64 bits, donc 8 pixels. Du fait de l'information bidimensionnelle d'une image, plusieurs solutions de regroupement de pixels sont possibles. En effet dans l'objectif de mieux résister à une compression aval ou de comprimer en même temps que le chiffrement, il nous semble intéressant de regrouper les pixels avec leurs voisins les plus proches, illustré figure 63.

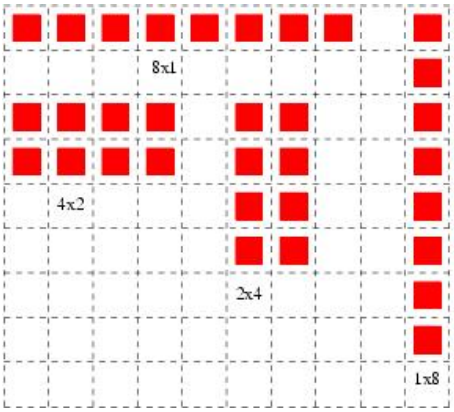


FIG. 63 – *Regroupements possibles des pixels dans l'image afin de former des blocs de 64 bits.*

Chaque bloc de 8 pixels sera crypté indépendamment par DES ou par TEA. Le bloc crypté obtenu viendra alors se substituer dans l'image au bloc original. Dans ce chapitre, le parcours de lecture des blocs sera exécuté uniquement de manière linéaire (scan line). Nous verrons, chapitre suivant, qu'il est souvent plus intéressant d'utiliser d'autres types de parcours (spirale, zig-zag, ...).

4.3.2 Cryptage d'images par RSA

La longueur des blocs cryptés par RSA doit être plus petite que la valeur de n de manière à conserver une réversibilité. Cependant, afin de conserver les valeurs des pixels supérieurs ou égaux à n, il faut bien souvent effectuer un échantillonnage [Borie 02a, Borie 02b]. Pour ne pas perdre trop d'information à l'échantillonnage, la valeur de n doit également être le plus proche possible de la valeur maximale du bloc à coder sans dépasser cette valeur. En effet si la valeur de n est plus grande que cette valeur maximale, le bloc chiffré risque de nécessiter un nombre de bits plus important que le bloc initial.
Pour considérer des clefs plus grandes, les pixels voisins formant un bloc sont regroupés

de la manière suivante :

$$\begin{aligned} bloc[i/k] &= p(i) + 256 * p(i+1) + 256^2 * p(i+2) \\ &\quad +... + 256^{k-1} * p(i+k-1), \end{aligned} \tag{68}$$

avec k étant le nombre de pixels pris en compte pour former le bloc.

Du fait de la diffusion des clefs publiques, les clefs doivent être codées sur au moins 512 bits pour assurer une réelle sécurité, soit 64 pixels. De part la manipulation de grands nombres au niveau des équations (64) et (65), cette solution impose un temps de calcul relativement important.

4.3.2.1 Une première implémentation de RSA

Dans une première implémentation de l'algorithme RSA appliqué aux images [Borie 02b], le programme générait automatiquement les clefs en fonction des nombres p et q et créait bloc par bloc l'image cryptée ou décryptée.

Il est à noter que dans le cryptage d'une image par RSA et dans le cas où l'on veut que le fichier crypté soit également une image, contrairement aux méthodes de chiffrement symétriques par blocs comme DES, nous aurons une légère dégradation de l'image originale et l'image cryptée et décryptée, bien que souvent invisible pour le SVH (Système Visuel Humain). Cette dégradation s'explique de la manière suivante :

Supposons que le cryptage s'effectue pixel par pixel (8 bits). Alors le nombre n devra être inférieur à 256 pour s'assurer que le résultat tienne également sur 8 bits. Par exemple le plus grand n possible sera 253, produit des deux nombres premiers 23 et 11. Mais dans ce cas, les valeurs des pixels cryptés et décryptés ne pourront dépasser 252. En effet, à cause du modulo 253 et afin de conserver une bijection entre les pixels clairs et les pixels cryptés, les niveaux de gris supérieurs ou égaux à 253 ne peuvent pas être conservés. Nous effectuons donc un échantillonnage de l'histogramme de l'image, ce qui nous vaut une perte de 3 niveaux de gris [Borie 02a].

Dans le cas d'un chiffrement pixel par pixel, la sécurité du RSA est loin d'être assurée. En effet dans le cas de cette première implémentation le problème de manipulation des grands nombres n'a pas été abordé. En effet la taille des types standards dans les langages de programmation étant limitée, les blocs de données à chiffrer n'atteignaient que la taille de deux pixels, à savoir 16 bits (nombres de l'ordre de 10^4), alors que même 1024 bits (10^{300}) n'offrent à RSA, si l'adversaire est puissant, qu'une sécurité relative. Il a donc fallu trouver un moyen de résoudre ce problème, à savoir créer une méthode de gestion personnelle des grands nombres.

4.3.2.2 RSA et les grands nombres

La méthode consiste à créer un nouveau type de nombres entiers, que nous avons appelé le type `grand`, une structure qui puisse représenter tout nombre entier positif de taille quelconque, dans un espace mémoire adapté et dynamique. L'unité de mémoire que nous avons choisie est l'octet (type `unsigned char`) car c'est la taille la plus naturelle pour le stockage en mémoire, et celle que nous estimons être la plus efficace en terme de temps de calculs.

Par exemple, un nombre compris entre 0 et 255 sera codé sur un seul octet, entre 256 et $(256^2 - 1)$ sur deux octets, ... entre 256^{n-1} et $(256^n - 1)$ sur n octets.

Mais surtout, pour que ces grands entiers puissent être manipulés, nous devons être en mesure d'y effectuer les opérations de base que sont la somme, la différence, le produit, le quotient, le reste de la division entière (modulo) et enfin la comparaison de deux nombres.

Dans tous les cas, la méthode consistera à adapter les opérations afin de les programmer non pas en base 10, mais en base 256.

Prenons par exemple l'addition de deux nombres en base b, en comparant la base 256 à la base 10 que nous connaissons bien (figure 64). On calcule d'abord la somme des chiffres de droite. On place dans le résultat la somme obtenue modulo b. Si cette somme est supérieure ou égale à b, alors on garde une retenue de 1 pour la somme des deux prochains chiffres.

```
  1 6 4 6 3          64  79
+ 1 1 9 8 2        + 46 206
    1 1 1 1          1
-----------        ---------
  2 8 4 4 5        111  29
```

FIG. 64 – *Addition de deux nombres en base 10 et en base 256.*

Il n'y a donc pas de réelle nouveauté dans la démarche à suivre, et ce pour n'importe quelle base $b \in \mathbf{N}^*$. Nous ne détaillerons donc pas les algorithmes de chaque opération, mais on imagine que la division par exemple (qui fait également office de *modulo*) est plus difficile à mettre en œuvre (figure 65).

Il ne faut pas oublier que chaque fonction opération se charge d'allouer la taille nécessaire en octets du nombre résultat, tout en le codant sur le nombre exact, et pas plus, d'octets dont il a besoin (par exemple, on ne code pas un nombre inférieur à 256 sur deux octets avec le premier à zéro).

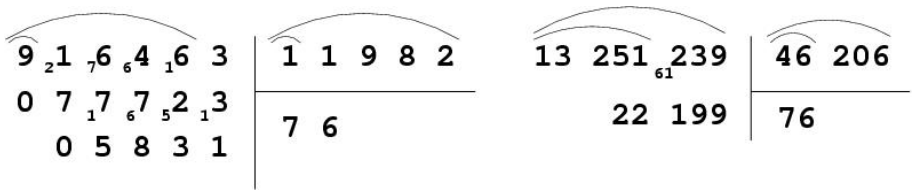


FIG. 65 – *Division de deux nombres en base 10 et en base 256.*

4.3.3 Chiffrement d'images par flot asynchrone

Dans cette section nous présentons un nouvel algorithme de chiffrement par flot asynchrone appliqué aux images. Soit K une clef de longueur k bits $b_\imath$ et $K = b_1 b_2 ... b_k$. L'unité de cryptage est le pixel (1 octet). La méthode réside dans le fait que pour chaque pixel de l'image le cryptage dépend du pixel original, de la valeur de la clef K, et des $k/2$ pixels précédemment cryptés. Pour utiliser les équations (67) nous avons $t = k/2$.

Pour chaque pixel $p_\imath$ de l'image originale, nous calculons la valeur du pixel $p'_\imath$ de l'image chiffrée en utilisant l'équation suivante :

$$\left\{ \begin{array}{rcl} z_\imath & = & (\sum_{\jmath=1}^{k/2} \alpha_\jmath p'_{\imath-\jmath}) mod\ 256 \\ p'_\imath & = & (z_\imath + p_\imath) mod\ 256, \end{array} \right. \tag{69}$$

avec $\imath \in [0, \ldots, N-1]$, où N est le nombre de pixels de l'image, k est la longueur de la clef, avec $k \in [1,N]$, et $\alpha_\jmath$ est une séquence de $k/2$ coefficients générée à partir de la clef secrète K [Puech 01c, Puech 01b].

Le principe de chiffrement est le même que celui illustré figure 61.a. Les équations (69) ont une récurrence d'ordre $k/2$, correspondant à la moitié de la longueur de la clef [Puech 01a, Puech 01d]. Les coefficients $\alpha_\jmath$ sont des coefficients entiers compris entre -2 et $+2$ tel que :

$$\left\{ \begin{array}{ll} \alpha_\jmath = \beta_\jmath - 1 & \text{si } \beta_\jmath \in \{0,1,2\}, \\ \alpha_\jmath = \pm 2 & \text{si } \beta_\jmath = 3, \end{array} \right. \tag{70}$$

avec $\beta_\jmath = 2b_{2\jmath-1} + b_{2\jmath}$, où $b_{2\jmath-1}$ et $b_{2\jmath}$ sont deux bits voisins de la clef secrète K.

De plus, la densité de probabilité des $\alpha_\jmath$ doit être uniforme afin d'atténuer le bruit durant l'étape de décryptage. Le signe devant les coefficients égaux à 2 dépend de la somme des coefficients $\alpha_\jmath$ afin d'avoir :

$$\frac{1}{k/2} \sum_{\jmath=1}^{k/2} \alpha_\jmath \simeq 0. \tag{71}$$

Une autre information est également construite à partir de la clef K. En effet, en prenant en compte que le chiffrement d'un pixel s'appuie sur les $k/2$ pixels précédemment cryptés,

nous ne pouvons pas chiffrer les $k/2$ premiers pixels de l'image de la même manière. Il est nécessaire d'associer la séquence des coefficients $\alpha_\imath$ à une séquence de $k/2$ pixels virtuels cryptés $p'_{-\imath}$, pour $\imath \in [1,\ldots,k/2]$, correspondant à un vecteur d'initialisation (VI). Par conséquent, un VI est codé dans la clef : $k/2$ valeurs de pixels virtuels qui permettent de crypter les $k/2$ premiers pixels de l'image comme si ils avaient des prédécesseurs.

La longueur k de la clef K doit être suffisamment grande afin de garantir une sécurité maximale. Supposons $k = 128$, comme nous avons 2 bits par coefficient α, l'ordre de la récurrence est 64. Concernant le VI, nous avons montré qu'il était nécessaire pour chiffrer les $k/2$ premiers pixels. Nous ne lui donnons donc pas plus de place supplémentaire dans la clef, mais la valeur du VI est déduite de la clef de 128 bits par le principe suivant. Il est basé sur une fenêtre glissante qui lit les bits de la clef de la gauche vers la droite. La fenêtre lit le premier octet afin de générer le premier pixel virtuel, le système se déplace alors d'un bit de la clef vers la droite afin d'obtenir un nouvel octet et de générer un autre pixel virtuel. Le déplacement de un bit vers la droite s'effectue jusqu'à obtenir le nombre nécessaire de pixels virtuels.

L'équation 72 présente la procédure de décryptage. Dans la procédure de décryptage, nous devons appliquer le processus inverse. Nous pouvons noter que la fonction génératrice la clef dynamique est la même qu'à l'équation (69):

$$\left\{ \begin{array}{rcl} z_\imath & = & (\sum_{\jmath=1}^{k/2} \alpha_\jmath p'_{\imath-\jmath}) mod\ 256 \\ p_\imath & = & (p'_\imath - z_\imath) mod\ 256, \end{array} \right. \tag{72}$$

4.4 Résultats

Dans cette section, nous appliquons les différents algorithmes de chiffrement à des images réelles.

4.4.1 Cryptage d'images par DES et TEA

A partir de la figure 66.a, nous avons appliqué l'algorithme DES par blocs de 8 pixels en ligne avec une clef de 64 bits pour obtenir l'image de la figure 66.b. De la même manière, avec l'algorithme TEA par blocs de 8 pixels en ligne avec une clef de 128 bits, nous obtenons l'image de la figure 66.c. Nous constatons que l'information initiale n'est plus du tout visible. En comparant l'histogramme de l'image initiale, figure 66.d avec les histogrammes des images cryptées, figure 66.e et f, nous constatons que les probabilités d'apparition des niveaux de gris sont équitablement réparties. Par conséquent, les entropies

des images cryptées sont très élevées (proches de 8 bits/pixel). De ce fait, les techniques de compression supprimant les redondances deviennent difficiles à appliquer de manière directe.

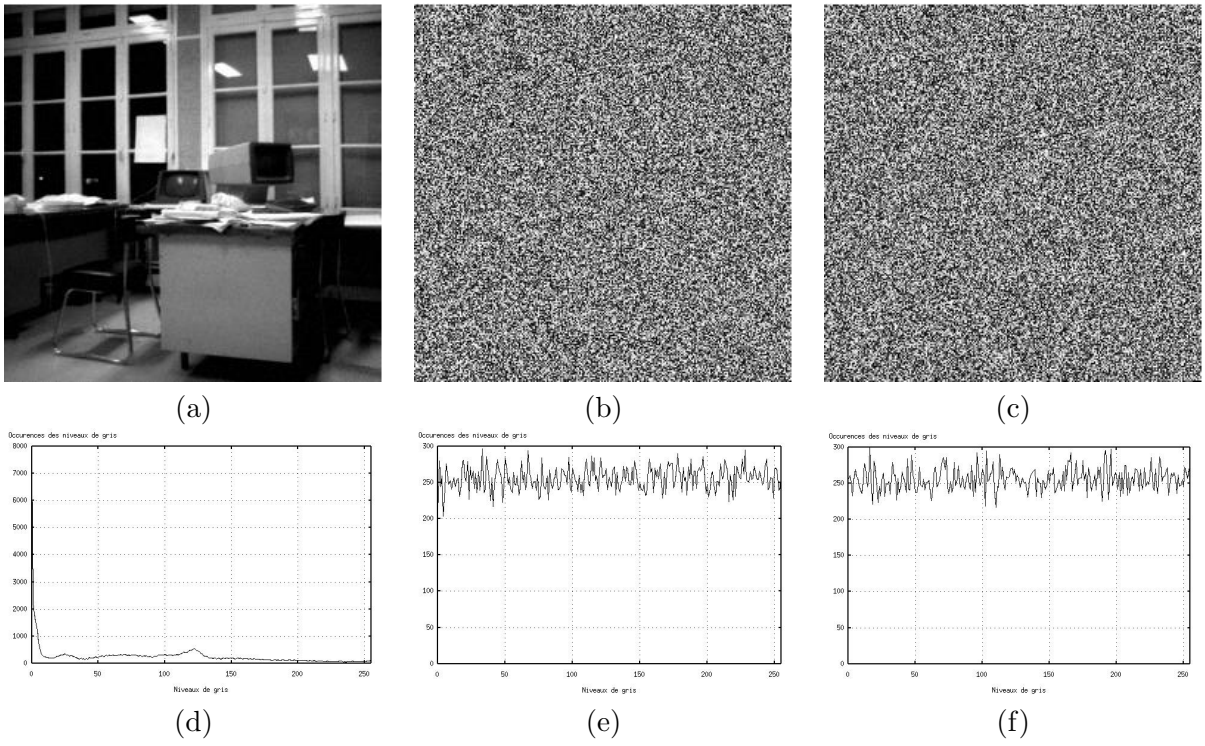


FIG. 66 – *Résultats de cryptage d'images par bloc, a) Image originale (65ko), b) Image cryptée par l'algorithme DES par bloc de 8 pixels en ligne avec une clef de 64 bits, c) Image cryptée par l'algorithme TEA par bloc de 8 pixels en ligne avec une clef de 128 bits, d), e) et f) Histogrammes correspondants.*

Dans le cas de la figure 67.a, apparaissent des zones homogènes (au niveau du ciel). En chiffrant cette image respectivement avec les algorithmes DES et TEA nous constatons, figures 67.b et c, l'apparition d'une texture. Cette texture provient du fait que deux blocs identiques dans l'image initiale donnent deux blocs cryptés identiques. Au niveau des histogrammes, figures 67.e et f, nous constatons la présence forte de 16 niveaux de gris correspondant au cryptage des niveaux de gris de deux zones homogènes. Les algorithmes de chiffrement par blocs posent problèmes dans le cas d'images contenant des zones homogènes.

Ce problème dû aux zones homogènes n'est pas sans conséquence lorsque nous testons nos programmes de cryptage des images médicales telle que celle de la figures 68.a. Sur les figures 68, nous voyons également apparaître sur les images cryptées des textures (sous

forme de rayures). Ces textures sont à l'origine de 8 pics de niveaux de gris sur les histogrammes figures 68.d et f. La raison de ce phénomène se trouve dans l'apparition de grandes zones homogènes (en l'occurence noires) sur les images médicales. En effet, nous avons vu précédemment que du fait que les algorithmes DES et TEA cryptent par blocs de 8 pixels, si plusieurs blocs identiques entre eux de valeur B_0 (contenant uniquement des pixels noirs) sont rencontrés, tous les B_0 seront cryptés de la même manière (B'_0), d'où la répétition des 8 niveaux de gris de B'_0 dans l'image cryptée. Le cryptage est alors très mauvais pour deux raisons : d'abord, parce que n'importe qui devine la nature de l'image (échographie), mais aussi parce que la connaissance de la valeur du bloc avant (on peut deviner que les pixels clairs de B_0 étaient tous noirs) et après cryptage (les 8 niveaux de gris dominants dans l'image cryptée) est un indice précieux pour les cryptanalystes.

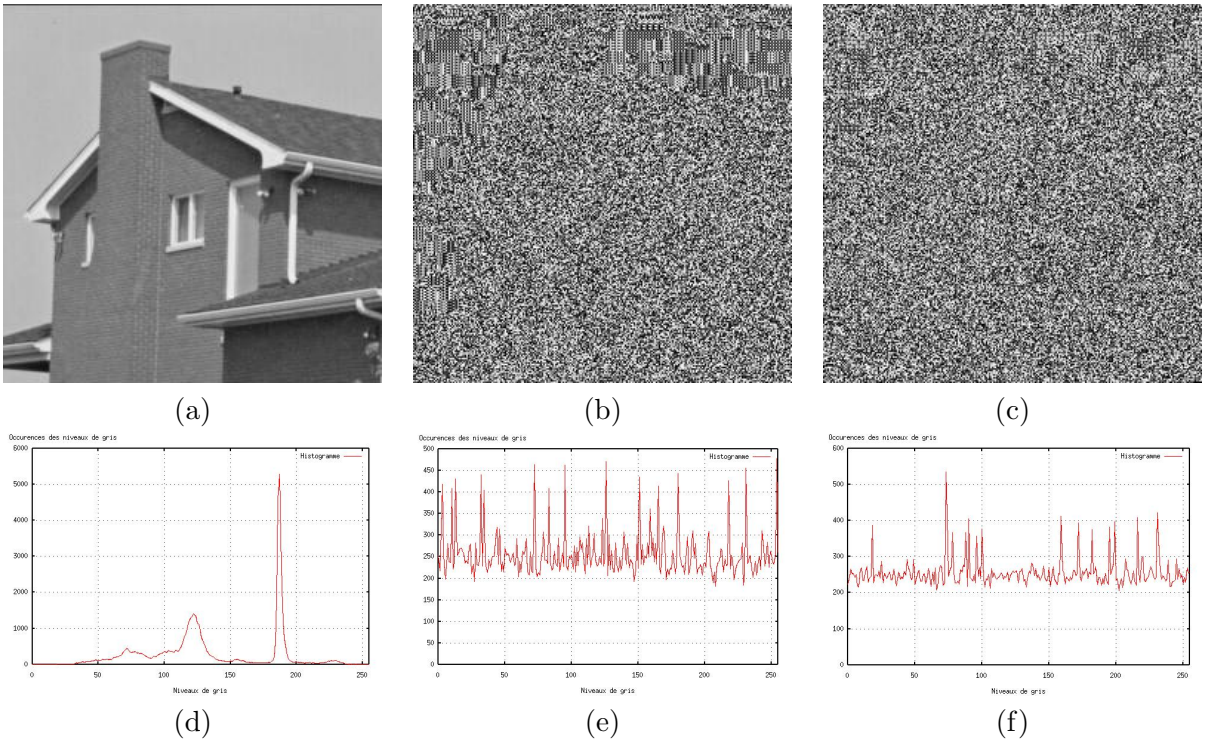


FIG. 67 – *a) Image originale, b) Image cryptée par l'algorithme DES par bloc de 8 pixels en ligne avec une clef de 64 bits, c) Image cryptée par l'algorithme TEA par bloc de 8 pixels en ligne avec une clef de 128 bits, d), e) et f) Histogrammes correspondants.*

Pour ce qui est des temps de calcul, l'algorithme DES, pourtant reconnu comme étant rapide, est nettement dépassé par l'algorithme TEA (figure 69)[3]. Alors que le DES nécessite entre 35 secondes (Pentium II 500 MHz, 256 Mo de RAM) pour crypter une

3. On notera que les mesures de temps ont été réalisées avec nos programmes et que leur justesse dépend du niveau d'optimisation des algorithmes.

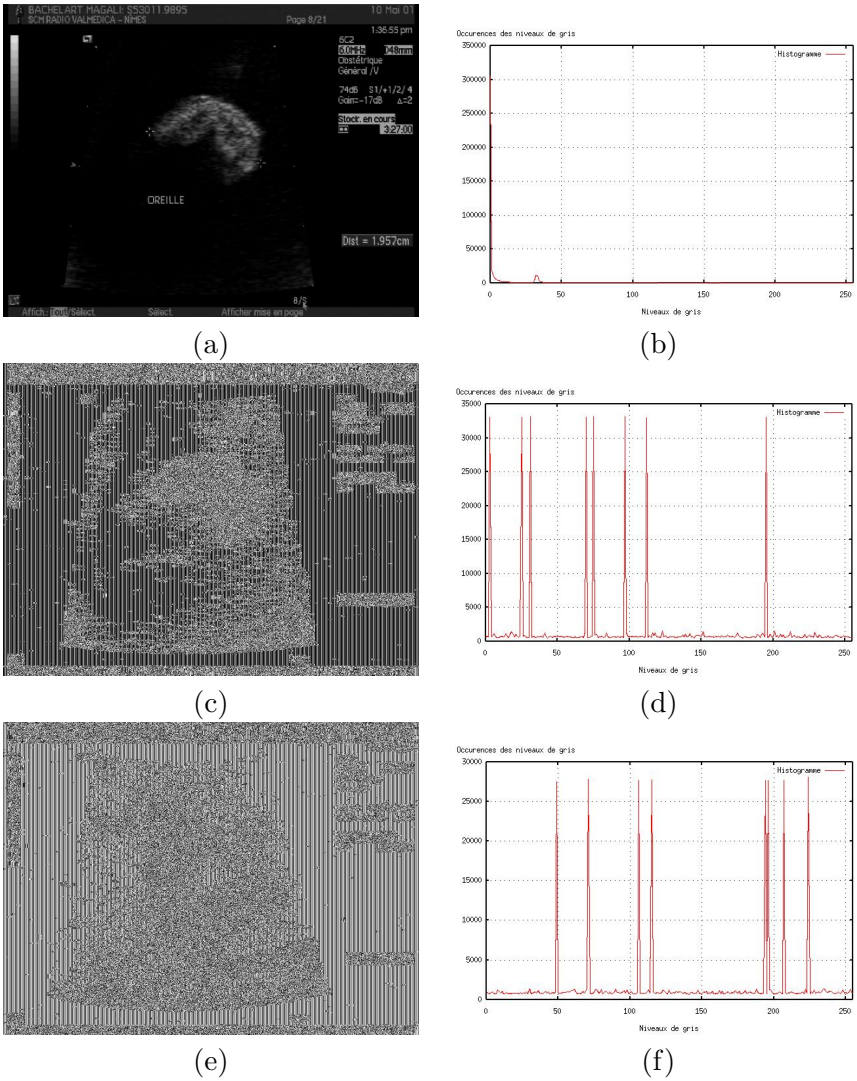


FIG. 68 – *a) Image médicale échographique (442 ko), avec de grandes zones homogènes, b) Image médicale cryptée par l'algorithme DES par bloc de 8 pixels avec une clef de 64 bits, c) Image médicale cryptée par l'algorithme TEA par bloc de 8 pixels avec une clef de 128 bits, d), e) et f) histogrammes correspondants.*

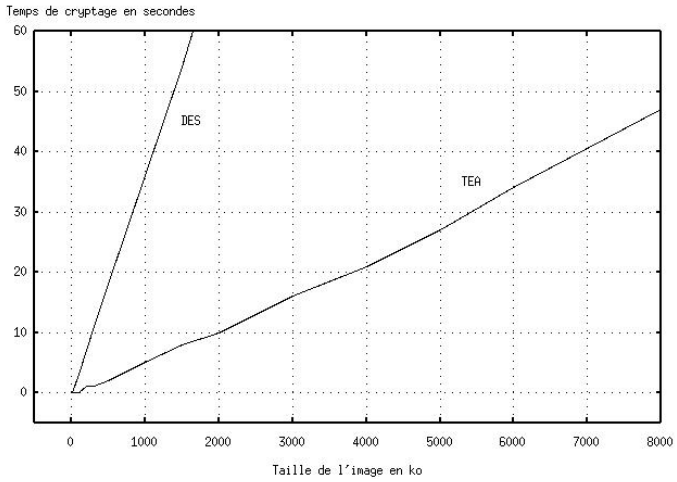


FIG. 69 – *Comparaison des vitesses de cryptage de DES et TEA.*

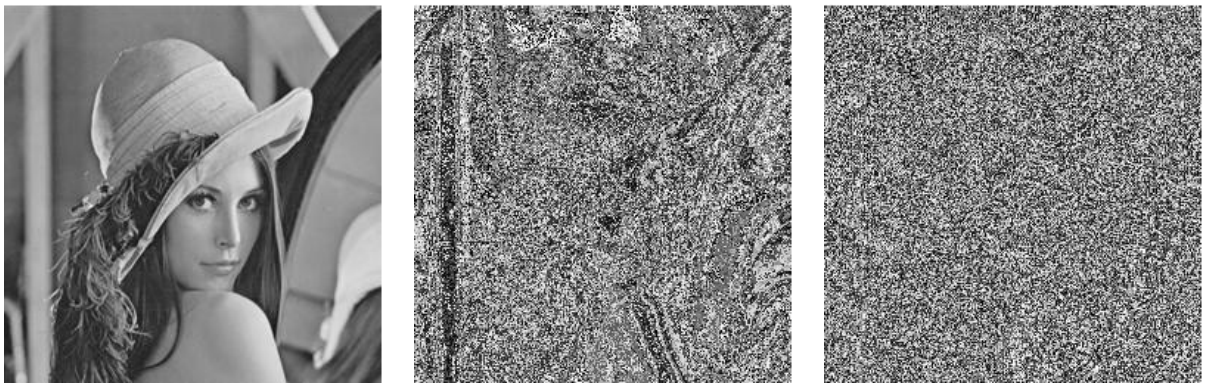
image de la taille de 1 Mo, le TEA crypte la même image en 5 secondes seulement. La rapidité de TEA rend alors réalisable le cryptage pour des images allant jusqu'à 10 Mo en moins d'une minute.

4.4.2 Cryptage d'images par RSA

Comme nous l'avons présenté dans la section 4.3.2, l'algorithme RSA est malheureusement ou soit pas assez sécurisé (si n est codé sur trop peu d'octets) ou trop lent. Dans notre cas, pour le cryptage d'images, souvent de gros fichiers, il n'y a pas vraiment de compromis entre les deux, cryptage et décryptage étant très souvent à la fois trop longs et trop facilement cassables. Néanmoins, voici quelques résultats que nous avons obtenus avec un cryptage rapide mais une sécurité quasi-inexistante (8 bits figure 70.b, ou 16 bits, figure 70.c), puis avec une sécurité un peu meilleure (64 bits), ce qui nécessite l'implémentation des grands nombres mais des temps de calcul très importants (figure 71).

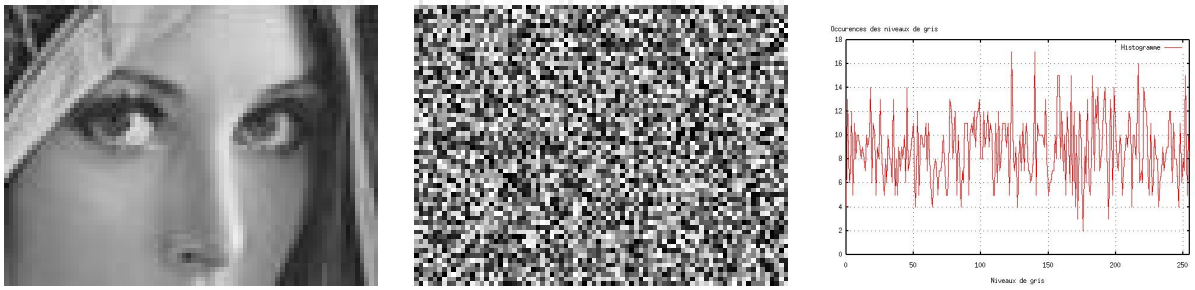
On notera que dans le cas du cryptage sur 8 bits figure 70.b, où les pixels sont cryptés un par un, on a affaire à ce qu'on appelle un chiffrement par substitution, où à chaque pixel (de 0 à 253) clair correspond un pixel crypté.

Afin d'appliquer l'algorithme RSA dans un temps convenable, nous avons dû prendre une image de petite taille 56×40 pixels, illustrée figure 71.a, pour obtenir l'image cryptée figure 71.b. La figure 71.c présente l'histogramme de l'image cryptée et nous permet de constater que l'entropie est également proche de 8 bits/pixel. Nous avons analysé le temps de cryptage en fonction de la taille des blocs, figure 72.a. Il faut déjà plus de 10 secondes



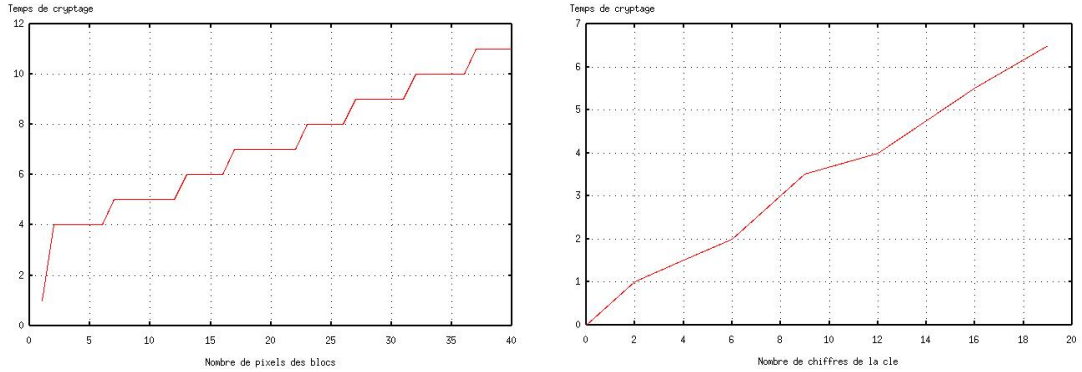
(a) (b) (c)

FIG. 70 – *a) Image originale (65 ko), b) Image originale cryptée par RSA avec n proche de 256 (8 bits), c) Image originale cryptée par RSA avec n proche de* 256^2 *(16 bits).*



(a) (b) (c)

FIG. 71 – *a) Image originale (sous-partie de la figure 70.a), 4 ko, b) Image (a) cryptée par RSA avec n proche de* 256^8 *(64 bits), c) Histogramme.*



(a) (b)

FIG. 72 – *a) Temps de cryptage en seconde par RSA en fonction du nombre de pixels dans les blocs de cryptage, b) Temps de cryptage en seconde par RSA en fonction de la longueur de la clef privée.*

pour crypter une image de 2240 pixels par blocs de 40 pixels. En fixant la valeur du n, nous avons également analysé le temps de cryptage en fonction de la valeur de e choisie entre 2 et $\phi(n)$, figure 72.b. Nous constatons qu'en prenant des petites valeurs de e le temps de cryptage semble intéressant. Malheureusement, en calculant l'inverse de e, il y a de fortes chances pour obtenir une grande valeur, donc un temps long de décryptage.

Nous sommes ainsi parvenus, grâce au codage des grands entiers, à adapter les types et les opérations au programme RSA existant pour pouvoir crypter sur plus de 2 pixels. Nous avons expérimenté jusqu'à 64 pixels, et les résultats se sont avérés corrects.

Malheureusement, le chiffrement (ou déchiffrement) RSA a un inconvénient majeur (en plus de celui de la dégradation dûe à l'échantillonnage), qui est celui de la lenteur d'exécution. L'ancien programme chiffrant par blocs de deux pixels était rapide, mais le fait de passer les nombres dans le type `grand` et d'effectuer les opérations sur de très grands nombres fait exploser les temps de calcul (figure 73) et nous avons donc dû faire nos tests sur des images très petites.

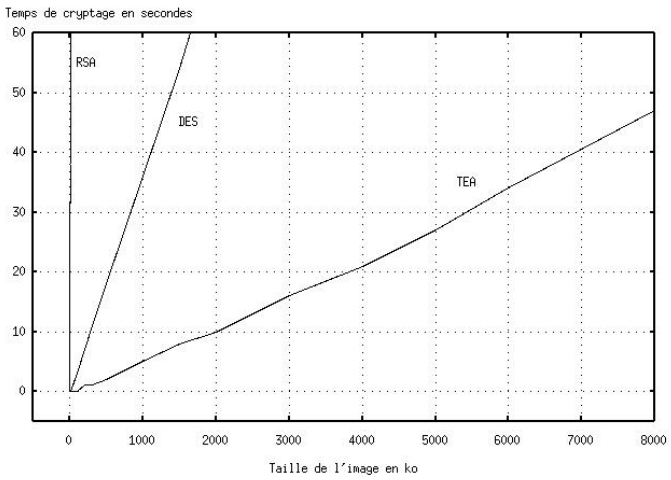


FIG. 73 – *Temps de cryptage de RSA comparé à DES et TEA.*

Toujours est-il que RSA est effectivement réputé pour être très coûteux en temps de calcul, de l'ordre de 1000 fois plus lent que DES [Schneier 97]. Nous avons jugé qu'il n'etait donc pas adapté pour le cryptage des données d'une image et nous préfèrerons un système symétrique comme DES ou TEA par exemple. En revanche, RSA ayant l'avantage d'être à clef publique, il peut être intéressant afin de transmettre à l'expéditeur de manière sécurisée la clef privée qui servira à déchiffrer le message crypté par DES ou TEA. Cette clef (par exemple de 128 bits) est suffisamment petite pour pouvoir être cryptée avec RSA

dans les plus brefs délais, même avec n très grand.

4.4.3 Cryptage d'images par flot asynchrone

Dans cette section, nous présentons les résultats de notre méthode de chiffrement par flot asynchrone. A partir de l'image oginale, figure 74.a (396×400 pixels), nous avons appliqué notre algorithme de chiffrement par flot avec une clef de 128 bits. La valeur de cette clef (en hexadécimal) est :
$K = 2B28AB097EAEF7CF15D2154F16A6883C$.
A partir de cette clef secrète, nous obtenons les valeurs de $\alpha(\imath)$ et $p(-\imath)$, illustrées tableau 18. Nous pouvons noté que $Pr(\alpha_\imath = 0) = \frac{14}{64} \simeq \frac{1}{4}$, $Pr(\alpha_\imath = \pm 1) = \frac{34}{64} \simeq \frac{1}{2}$ et $Pr_\alpha(\imath = \pm 2) = \frac{16}{64} \simeq \frac{1}{4}$. A partir des équations (71) nous avons $\frac{1}{k/2} \sum_{\jmath=1}^{k/2} \alpha_\jmath = 0.0625 \simeq 0$.

La figure 74.c illustre les valeurs obtenues pour la clef dynamique $z_\imath$ générée par notre méthode à partir de l'équation (69). Nous pouvons remarqué, figure 74.d, que la probabilité d'apparition de chaque valeur est quasi-uniforme. Donc la fonction génératrice de la clef dynamique $g()$ produit une séquence avec une grande période et de bonnes propriétés statistiques qui peut être appelé séquence binaire pseudo aléatoire (SBPA).

A partir des équations (69) nous obtenons l'image cryptée figure 74.e, nous remarquons que l'image initiale n'est plus du tout visible. En comparant l'histogramme, figure 74.b, de l'image initiale avec l'histogramme, figure 74.f, de l'image cryptée, nous remarquons que la densité de probabilité des niveaux de gris est quasi-uniforme. Par conséquence, les entropies des images cryptées sont très élevées (proche de 8 bits/pixel).

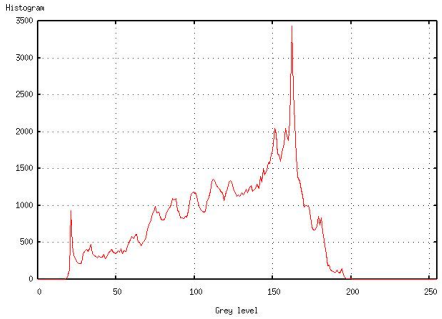
$\alpha(\imath)$	-1 1 1 2 -1 1 1 -1 1 1 1 -2 -1 -1 1 0 0 2 -2 1 1 1 2 1 -2 2 0 -2 2 -1 -2 2 -1 0 0 0 -2 0 -1 1 -1 0 0 0 0 -1 2 -2 -1 0 0 1 1 1 0 1 1 -1 1 -1 -1 2 -2 -1
$p(-\imath)$	43 86 172 89 178 101 202 148 40 81 162 69 138 21 42 85 171 86 172 88 176 97 194 132 9 18 37 75 151 47 95 191 126 253 250 245 234 213 171 87 174 93 187 119 239 222 189 123 247 239 223 190 124 249 243 231 207 158 60 120 241 226 197 138

TAB. 18 – *Valeurs de $\alpha(\imath)$ et $p(-\imath)$.*

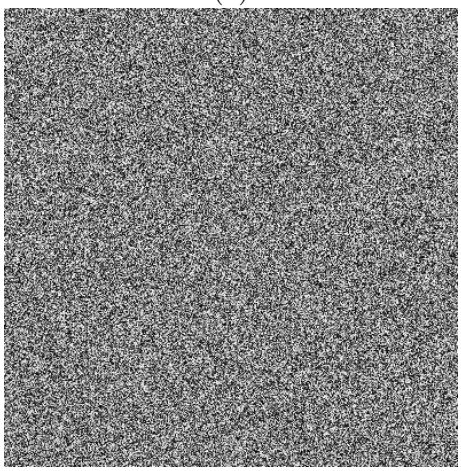
Afin de comparer notre méthode de chiffrement par flot asynchrone aux méthodes de chiffrement par bloc présentées précédemment nous avons appliqué notre algorithme de chiffrement par flot avec une clef de 64 bits sur les images bureau, figure 66.a , maison, figure 67, et échographique, figure 68.

Des images telles que la plupart des photographies par exemple donnent après cryptage des résultats aussi bons que ceux de DES ou TEA. En revanche, le chiffrement par flots

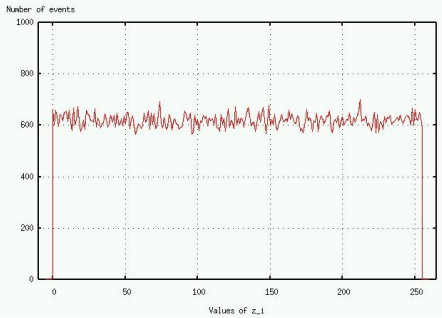
(a)



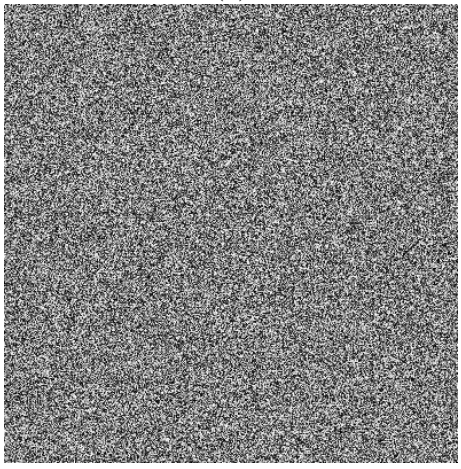
(b)



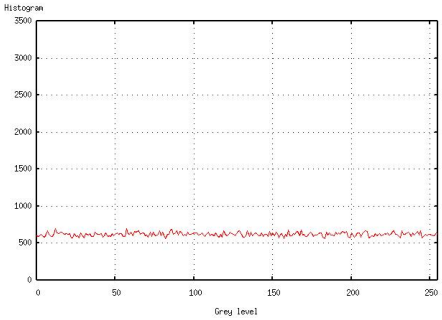
(c)



(d)



(e)



(f)

FIG. 74 – *a) Image originale, b) Histogramme de l'image originale, c) Image de la clef dynamique z_i, d) Histogramme des valeurs de la clef dynamique z_i, e) Image finale cryptée avec l'algorithme de chiffrement par flot asynchrone, avec une clef de 128 bits, f) Histogramme de l'image cryptée.*

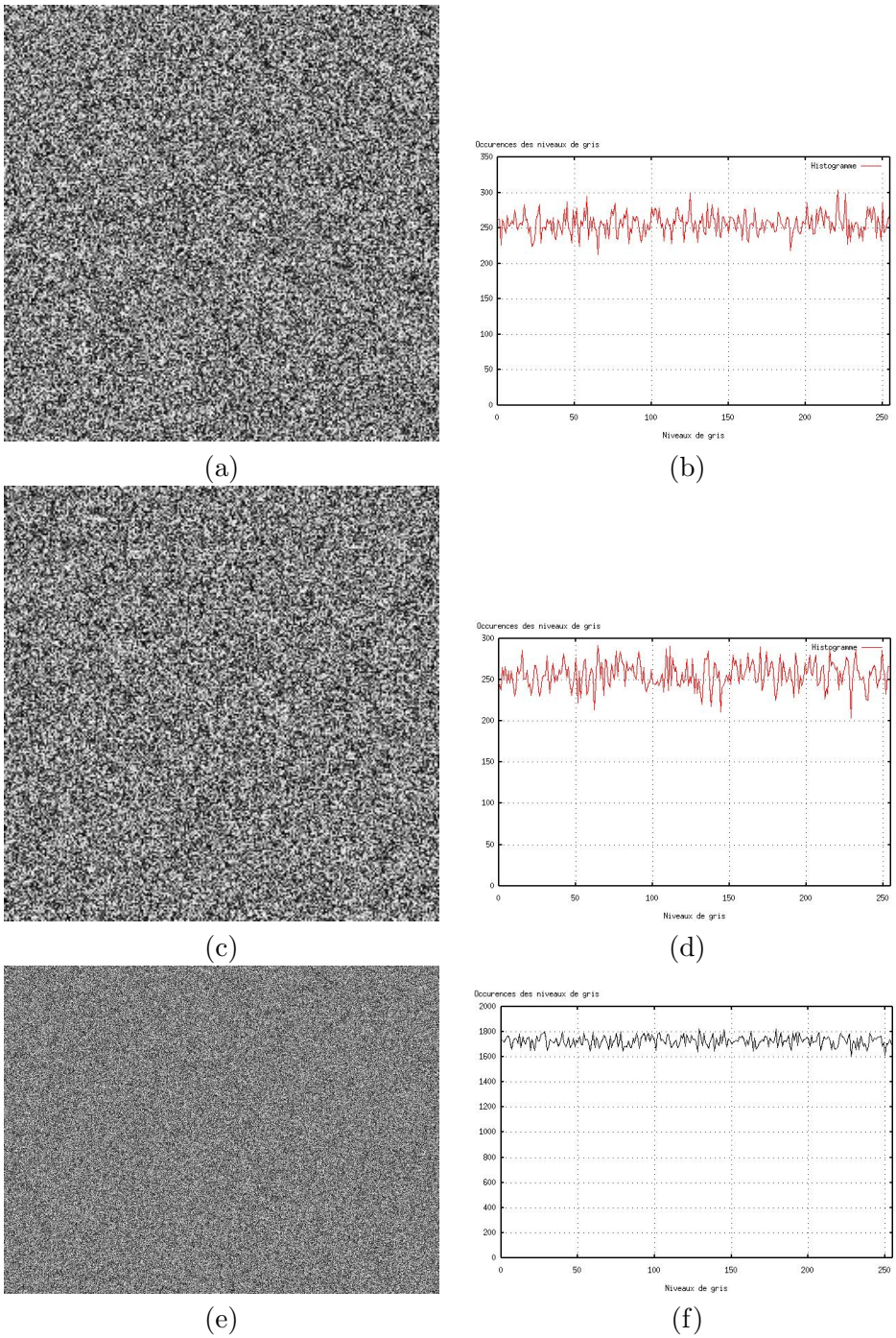


FIG. 75 – *Algorithme de chiffrement par flot asynchrone, avec une clef de 64 bits a) Image bureau, figure 66.a, cryptée, b) Histogramme de l'image (a) cryptée, c) Image maison, figure 66.a, cryptée, b) Histogramme de l'image (c) cryptée, e) Image échographique, figure 66.a, cryptée, b) Histogramme de l'image (e) cryptée.*

présente un avantage majeur par rapport aux autres systèmes de cryptage utilisés pour ce qui est des images médicales. En effet, puisqu'on tient compte pour chaque pixel à crypter du résultat du cryptage des pixels précédents, le problème des zones homogènes est résolu. Nous ne sommes plus dans le cas des systèmes de chiffrement par blocs où deux blocs clairs identiques donnaient le même bloc crypté.

Nous constatons que quelque soit le type d'image (avec ou sans zones homogènes), aucune texture n'apparaît dans les images cryptées. Nous constatons le même point au niveau des histogrammes des trois images cryptées. En conclusion dans le cas d'un chiffrement par flot, les zones homogènes ne sont plus visibles ni au niveau de l'image, ni au niveau de l'histogramme.

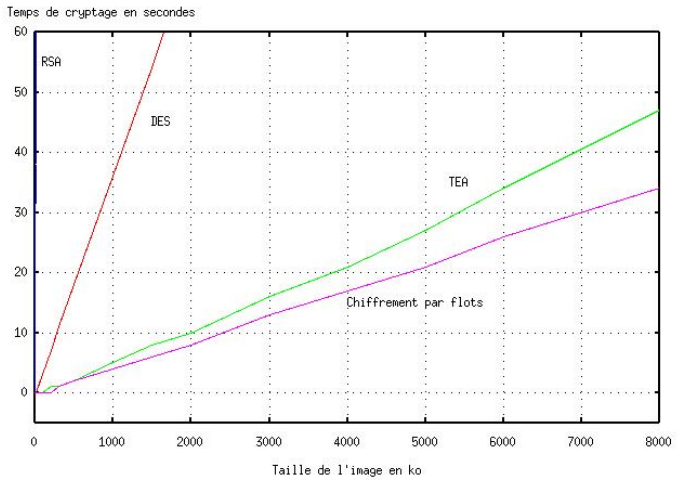


FIG. 76 – *Temps de cryptage en seconde pour chacune des 4 méthodes en fonction de la taille de l'image.*

De plus, notre méthode de chiffrement par flots présente un autre avantage : comme les calculs qui la composent sont peu nombreux, elle s'avère très rapide, plus encore que TEA. Désormais, une image de 7 Mo est cryptée en 30 secondes (figure 76).

La figure 76 illustre les temps de cryptage pour chacune des 4 méthodes en fonction de la taille de l'image (la courbe pour RSA est confondue avec l'axe des ordonnées). Les deux algorithmes intéressants au niveau du temps de calcul sont l'algorithme TEA et le chiffrement par flot. Il faut à peu près 20 secondes pour chiffrer une image de 4 Moctets avec ces deux algorithmes.

4.5 Compression d'images cryptées

Pour le transfert des images, un point important et problématique n'a pas été soulevé dans les sections 4.3 et 4.4. En effet, les images au format brut peuvent atteindre plusieurs centaines de Mo. De plus, les médecins par exemple ont souvent besoin d'envoyer une grande séquence d'images. Dans les sections précédentes, nous avons mis l'accent sur les temps de cryptage et de décryptage de nos systèmes, mais pour pouvoir être transmises plus rapidement sur le réseau il est important que les images soient comprimées.

Un des objectifs de nos travaux est du faire du codage conjoint cryptage et compression. Bien évidemment, la solution la plus intuitive serait d'effectuer une compression de l'image, puis de crypter le tout avant de l'envoyer. Mais cela irait à l'encontre de notre optique. En effet, une image comprimée n'est plus une image mais un flux binaire de données. Par conséquent le cryptage qui s'ensuit est du cryptage de données binaires classique, mais n'est plus du cryptage d'images comme nous l'avons étudié sections 4.3 et 4.4. Avant de développer des méthodes de codage conjoint cryptage et compression, nous avons souhaité faire une analyse de la robustesse à la compression de nos méthodes de cryptage d'images.

Dans le cadre de cette section, nous effectuons le cryptage d'abord, et comprimons ensuite l'image cryptée avant de l'envoyer sur le réseau. A la réception, on décomprime l'image, puis on la décrypte.

Quelque soit le type d'images et quelque soit l'algorithme de chiffrement utilisé, nous avons constaté, section 4.4, par l'intermédiaire des histogrammes que l'entropie des images cryptées étaient proches de sa valeur maximale (8 bits/pixel). De ce fait, il n'y a plus de redondance dans les images cryptées, redondance sur laquelle s'appuient les algorithmes de compression.

Le résultat est immédiat. Avec l'algorithme JPEG et pour un facteur de qualité (FQ) de 100%, l'image cryptée augmente de taille lors de la compression. De plus, un autre problème encore plus important survient suite à notre démarche, à savoir celui de la perte due à la compression. Pour une image en clair, les pertes subies par l'image sont souvent imperceptibles. Par contre, la moindre modification d'un pixel de l'image cryptée entraîne fatalement la perte de tout le bloc (8 pixels) qui ne peut plus être décrypté.

Pour illustrer ces problèmes, voici les résultats obtenus après décompression puis décryptage des images en utilisant les systèmes de cryptage tels que nous les avons présentés.

4.5.1 DES et TEA

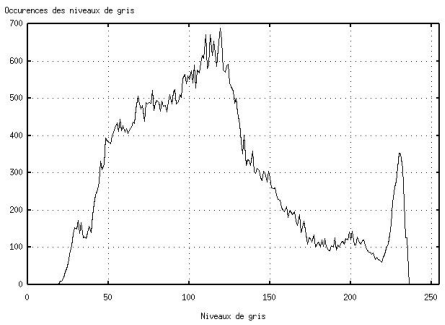
Les algorithmes DES et TEA sont assez comparables compte tenu du fait qu'ils cryptent par bloc de 64 bits, et que la modification d'un bit du bloc crypté ne permet plus le décryptage du bloc courant. Les résultats entre le DES et le TEA sont similaires et pour cette raison nous présentons dans les figures 77 que les résultats obtenus avec l'algorithme DES. A partir de l'image originale, figure 77.a, nous avons appliqué l'algorithme DES par bloc de 64 bits, puis comprimé et décomprimé avec JPEG l'image cryptée. Avec un FQ de 100% le décryptage de l'image décomprimée est illustré figure 77.c. Nous remarquons que l'image n'est presque plus visible. Cependant la forme de l'histogramme de l'image décryptée, 77.d, garde l'allure de l'histogramme de l'image originale, figure 77.b. Par contre, si l'on utilise un facteur de qualité de 90%, alors l'image décryptée, illustrée figure 77.e n'est plus du tout exploitable. Nous remarquons que dans ce cas l'histogramme associé, illustré figure 77.f, est relativement plat, donc l'image décryptée est une image aléatoire.

Nous ne présentons pas de résultats concernant l'algorithme RSA du fait que celui-ci n'est pas adapté au cryptage d'images, mais aussi parce que tel que nous l'avons testé, il construit également des blocs de 64 bits, donc les résultats sont similaires à l'algorithme DES.

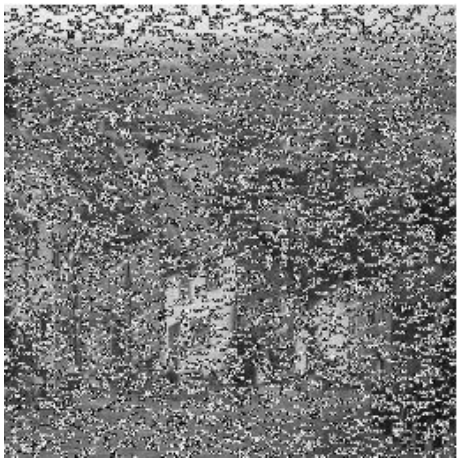
4.5.2 Chiffrement par flot

En revanche, l'algorithme de chiffrement par flot, présenté section 4.3.3, sans aucun traitement particulier, présente un intérêt certain par rapport à la compression. Etant composé d'opérations linéaires utilisant des petits coefficients, de faibles dégradations dans l'image cryptée due à la compression vont pouvoir être plus ou moins récupérables après l'étape de décryptage. Pour un facteur de qualité de 100%, le SVH (système visuel humain) ne distingue pas sinon peu de différences entre l'image de départ et celle qui est passée par le cryptage et la compression. Pour des FQ légèrement plus bas, les erreurs apparaissent mais les images sont nettement mieux reconstituées qu'avec un chiffrement par bloc. Les résultats sont présentés sur les figures 78. A partir de l'image originale, figure 77.a, nous avons appliqué l'algorithme de chiffrement par flot, puis comprimé et décomprimé avec JPEG l'image cryptée. Avec un FQ de 100% le décryptage de l'image décomprimée est illustré figure 78.a. Nous remarquons que l'image est de très bonne qualité et que l'histogramme, figure 78.b, reste très proche de l'histogramme de l'image originale. Pour un

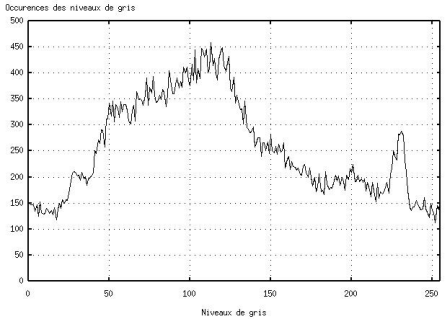
(a)



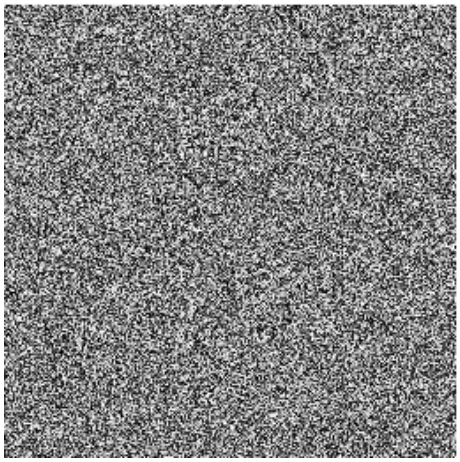
(b)



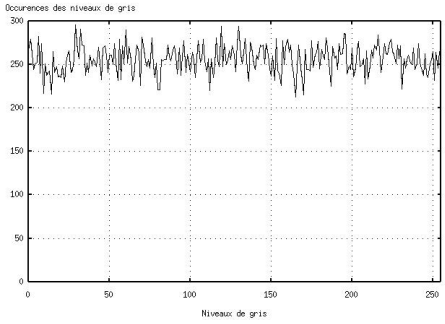
(c)



(d)



(e)



(f)

FIG. 77 – *Chiffrement DES et compression JPEG a) et b) Image originale et son histogramme, c) Image cryptée, puis compression/décompression de l'image cryptée avec un FQ de 100% suivi du décryptage, d) Histogramme de (c), e) Image cryptée, puis compression/décompression de l'image cryptée avec un FQ de 90% suivi du décryptage, f) Histogramme de (e).*

FQ de 90%, l'image décryptée illustrée figure 78.c, contient du bruit mais l'image reste reconnaissable. Dans ce cas, nous remarquons que l'histogramme, figure 78.d, a été lissé pendant la phase de décryptage. Pour un FQ de 80%, l'image décryptée illustrée figure 78.e, est encore plus bruitée que la précédente, mais elle reste de meilleure qualité que l'image de la figure 77.e chiffré par l'algortihme DES. Dans ce cas, nous remarquons que l'histogramme, figure 78.f, s'éloigne de l'histogramme de l'image originale.

Deux facteurs influent sur la robustesse à la compression de notre chiffrement par flot. Le premier est le nombre k de pixels précédemment cryptés à prendre en compte. Plus ce nombre est grand, et plus les erreurs se répercuteront. En effet, l'erreur survenue lors du cryptage d'un pixel sera combinaison linéaire des erreurs des k pixels cryptés précédents. Vu sous un autre angle, une erreur de cryptage sur un pixel altérera le cryptage des k pixels suivants. Mais comme nous l'avons souligné, c'est dans le nombre de coefficients que repose la sécurité du système. Afin de garantir une sécurité face aux attaques brutale, nous avons choisi $k = 64$.

Le second facteur important concernant la robustesse à la compression réside dans le choix des k coefficients α_i. Ils sont chacun codés sur 2 bits, formant une clef de $2k$ bits. Il est évident que pour que les erreurs se répercutent avec le minimum d'amplitude il est préférable de choisir les α_i proches de zéro. Mais en plus de cela, il est judicieux de faire en sorte que la somme des coefficients ne s'éloigne pas trop non plus de zéro.

4.5.3 Augmentation de la robustesse à la compression des images cryptées par bloc

Dans cette section nous proposons une nouvelle idée de cryptage robuste à la compression par bloc, plus basée sur une étude bidimensionnelle de l'image.

L'idée repose sur un découpage de l'image en blocs carrés de 3×3 pixels, soit 9 octets. 8 des 9 octets sont alors cryptés par l'algorithme TEA [4], tandis que le neuvième reste en clair. A la réception, on s'appuie sur le niveau de gris resté en clair pour essayer, malgré les pertes dues à la compression, de récupérer les 8 autres pixels.

Nous venons de voir que les algorithmes de chiffrement par blocs sont très sensibles aux erreurs rencontrées pendant le transfert des données ou dûes à une compression avec pertes. En effet, les algorithmes de compression d'images avec pertes se basent sur la suppression de redondances dans les images non détectables par le SVH. De ce fait, même

4. Nous avons effectué nos tests avec TEA du fait de sa rapidité mais il serait possible de choisir DES ou un autre système

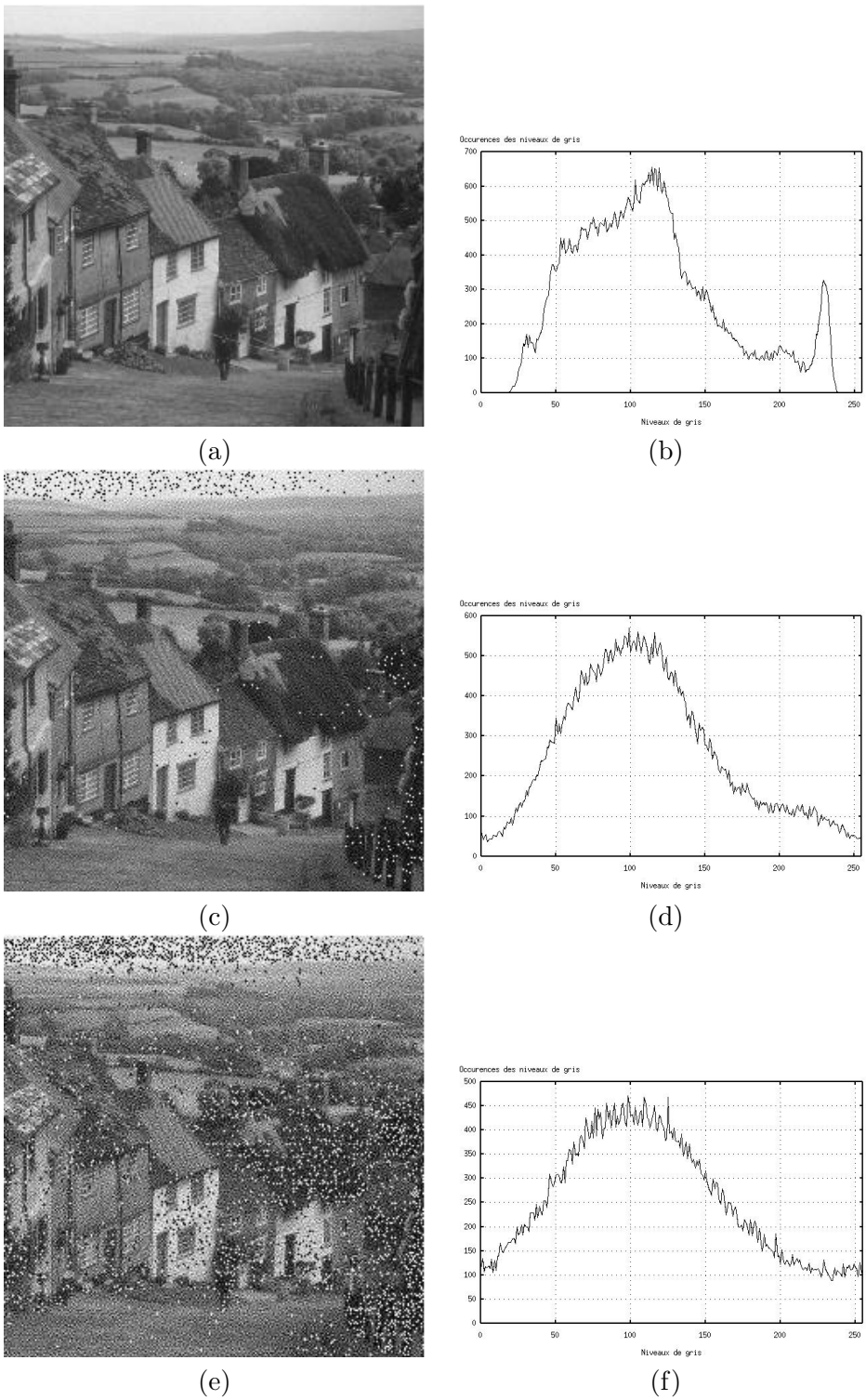


FIG. 78 – *Chiffrement par flot et compression JPEG a) Image cryptée, puis compression/décompression de l'image cryptée avec un FQ de 100% suivi du décryptage, b) Histogramme de (a), c) Image cryptée, puis compression/décompression de l'image cryptée avec un FQ de 90% suivi du décryptage, d) Histogramme de (c), e) Image cryptée, puis compression/décompression de l'image cryptée avec un FQ de 80% suivi du décryptage, f) Histogramme de (f).*

si un seul des 64 bits d'un bloc chiffré est erroné alors ce bloc déchiffré ne correspondra plus du tout au bloc initial. Afin de résister aux algorithmes de compression avec pertes, nous avons formé des blocs de 9 pixels (3×3) dont un des 9 n'est pas crypté par un algorithme par blocs mais par un algorithme de masquage robuste au bruit. Le positionnement du pixel masqué dépend de la clef de cryptage. Ces pixels plus robustes à la compression nous permettent de reconstruire à la réception une image basse résolution. Nous pouvons alors nous appuyer sur ces points pour détecter les éventuelles erreurs qu'ont subies les bits de poids faible des pixels cryptés.

4.5.3.1 Positionnement du pixel clair

Dans chaque bloc, le pixel clair devient non visible pour le SVH du fait qu'il est entouré d'une majorité de pixels cryptés. En revanche, si l'on connaît sa position dans le bloc 3×3 on peut récupérer l'image claire en basse résolution (1 pixel sur 9). Pour préserver la confidentialité de l'image il est donc nécessaire que la position des pixels clairs dans chaque bloc soit secrète et donc dépende de la clef, et varie d'un bloc à l'autre comme illustré figure 79.

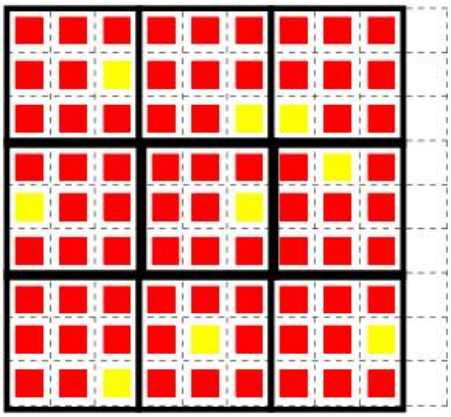


FIG. 79 – *Positionnement des pixels clairs dans les blocs de 9 pixels dont 8 sont chiffrés.*

La fonction qui pour chaque bloc va définir la position du pixel clair est la suivante. On note le numéro du bloc que l'on crypte (indice i incrémentant de 1 en 1 à chaque bloc), et on crypte ce nombre i codé sur 64 bits par TEA. On récupère un octet du nombre crypté obtenu (le premier par exemple), et on le projette dans l'intervalle $[0,8] \subset \mathbb{N}$ par l'opération $mod(9)$. On obtient donc un nombre entier compris entre 0 et 8 qu'on pourrait qualifier de nombre aléatoire recalculable grâce à la clef. De cette manière, sans décrypter l'image entière, le récepteur qui est en possession de la clef peut récupérer les niveaux de

gris d'un pixel sur neuf de l'image claire et de cette manière obtenir un aperçu de l'image décryptée en basse résolution.

4.5.3.2 Masquage du pixel clair

Afin de résoudre le problème du pixel clair nous devons essayer de le masquer, donc de crypter ce pixel clair de manière à ce qu'il soit possible d'en récupérer le niveau de gris même si sur l'image cryptée il a été légèrement modifié lors de la compression.

Pour déterminer la position du pixel clair dans le bloc, nous avons utilisé auparavant un nombre (indice de bloc) crypté par TEA dont nous avons utilisé le premier des 8 octets. Pour le masquage de ce pixel $p \rightarrow p'$, nous allons réutiliser ce nombre crypté, en prenant cette fois-ci le deuxième octet o_2 par exemple. Le pixel crypté p' aura alors la valeur $p' = p + o_2$.

A la réception, il suffira de reprendre l'équation dans le sens inverse, ce qui nous donne $p = p' - o_2$.

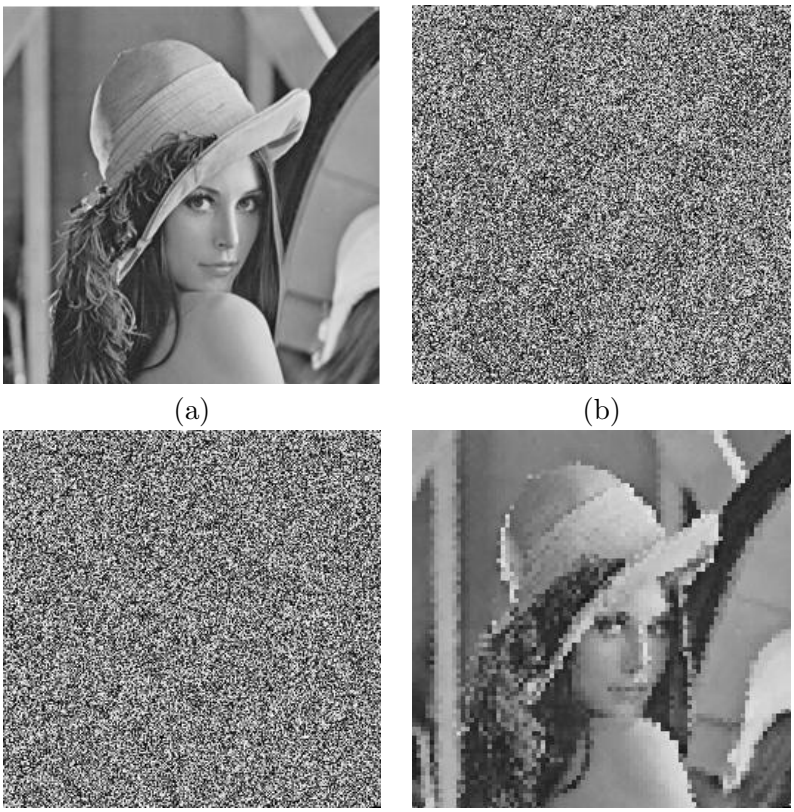
On peut donc, tout en récupérant le niveau de gris concerné, le crypter séparément par une opération robuste à la compression. Ainsi, l'histogramme de l'image cryptée ne contient plus la trace de celui de l'image claire

Afin d'illustrer cette méthode, nous avons utilisé l'algorithme TEA par bloc de 9 pixels contenant 1 pixel chiffré de manière différente (appelé pixel clair masqué). A partir de l'image originale de la figure 80.a nous avons appliqué l'algorithme TEA par blocs de 9 pixels en laissant 1 pixel en clair. L'information initiale subsiste encore dans l'image cryptée, figure 80.b, et également au niveau de l'histogramme, figure 81.a. Si nous appliquons au niveau de ces pixels clairs, une transformation plus robuste à la compression, nous obtenons l'image cryptée, figure 80.b et l'histogramme figure 81.c, où l'information initiale a complètement disparu.

Si l'on extrait, à l'aide d'un masque, uniquement les pixels clairs (masqués ou non) des images cryptées, nous obtenons une image basse résolution illustrée figure 80.d. Cette image basse résolution peut servir à augmenter la robustesse à la compression des algorithmes de chiffrement par blocs.

4.5.3.3 Application à la robustesse à la compression

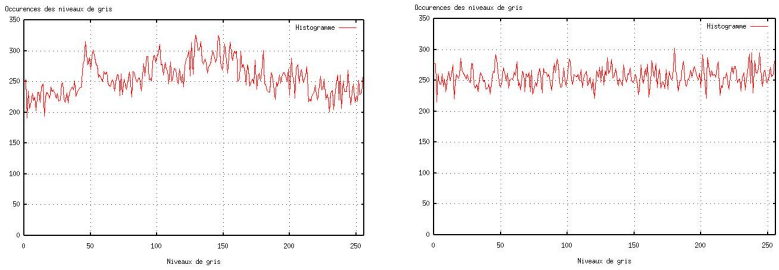
Jusque là, nous n'avons pas un système robuste à la compression si ce n'est pour un pixel sur neuf. Les autres sont cryptés par blocs de 8 et la moindre modification de l'un d'eux entraîne la perte de tout le bloc.



(a) (b)

(c) (d)

FIG. 80 – *a) Image originale, b) Image cryptée par l'algorithme TEA par blocs de 9 pixels dont 1 pixel en clair, clef de 128 bits, c) Image cryptée par l'algorithme TEA par blocs de 9 pixels dont 1 pixel en clair masqué, clef de 128 bits d) Image basse résolution reconstruite à partir des pixels en clairs.*



(a) (b)

FIG. 81 – *a) Histogramme de l'image cryptée par l'algorithme TEA par blocs de 9 pixels dont 1 pixel en clair, b) Histogramme de l'image cryptée par l'algorithme TEA par blocs de 9 pixels dont 1 pixel en clair masqué.*

L'idée est donc la suivante. Connaissant pour chaque bloc 3×3 la valeur d'un pixel, nous voudrions nous appuyer sur ce pixel pour essayer d'en déterminer les autres. La solution éventuelle serait de retrouver les valeurs de ces huit pixels cryptés telles qu'elles étaient avant compression, et de décrypter le bloc correspondant. Pour cela il nous faudrait étaler les combinaisons possibles pour ces pixels cryptés en tenant compte des pertes ayant pu avoir lieu selon le facteur de qualité de JPEG. Par exemple, si le FQ est de 100%, nous aurons généralement peu de pixels modifiés, et ceux qui auront été modifiés auront gagné ou perdu une seule unité de niveau de gris dans la majorité des cas. Un tel heuristique réduirait fortement le nombre de combinaisons à tester.

En revanche, si nous souhaitons effectuer une compression efficace sur les images cryptées, nous devrions nécessairement diminuer le facteur de qualité et dans ce cas, les lois combinatoires feraient exploser les temps de calcul.

A partir de l'image cryptée par l'algorithme TEA par blocs de 9 pixels dont 1 pixel clair masqué, figure 80.c, nous avons appliqué une compression JPEG avec deux facteurs de qualité différents. En comprimant l'image avec un facteur de qualité (FQ), $FQ = 100\%$, figure 82.a, après décompression puis décryptage, nous obtenons l'image de la figure 82.c dans laquelle un bon nombre de blocs 3×3 ne correspond plus aux données initiales. Par contre, toujours à partir de l'image figure 82.a il nous est possible de reconstruire l'image basse résolution, illustrée figure 82.e. Dans le cas où nous diminuons le facteur de qualité à $FQ = 80\%$, figure 82.b, l'image décomprimée, puis décryptée, figure 82.d ne nous permet plus de lire l'information initiale. Il est tout de même possible d'obtenir une image basse résolution, figure 82.f.

4.6 Une première approche de crypto-compression d'images

4.6.1 Méthode proposée

Dans le cas de certaines applications, comme l'imagerie médicale, les images initiales contiennent de grandes zones homogènes. Ces zones homogènes perturbent le cryptage d'images par blocs. En effet un bloc homogène est toujours chiffré de la même manière, ce qui entraîne l'apparition de textures dans l'image cryptée.

Pour résoudre ce problème, nous proposons d'analyser le contenu de l'image en même temps que le cryptage par blocs [Borie 04b, Borie 04a]. Si le bloc à chiffrer est homogène (8 pixels noirs par exemple) alors nous créons un bloc particulier qui va prendre en compte la série de blocs consécutifs homogènes. De ce fait, K blocs consécutifs identiques sont

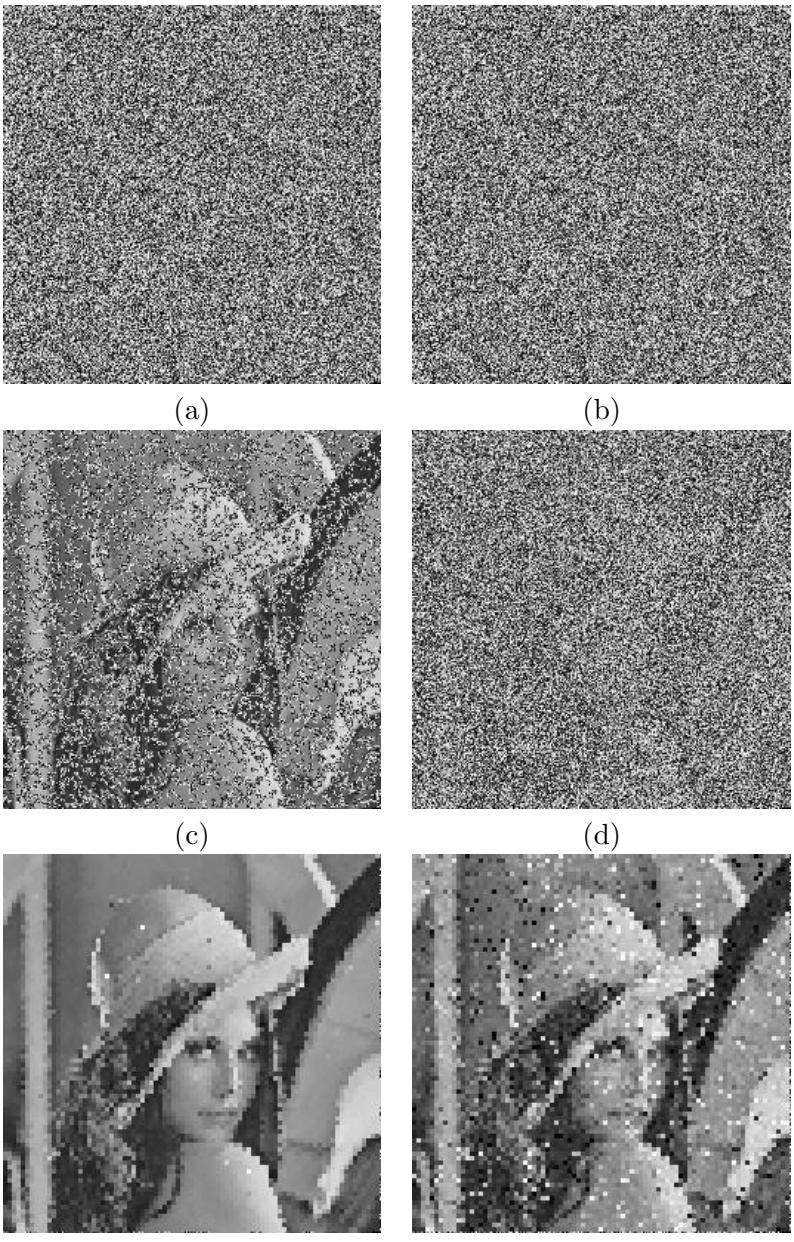


FIG. 82 – *a) Image cryptée par l'algorithme TEA par blocs de 9 pixels dont 1 pixel en clair masqué, puis comprimée avec un $FQ = 100\%$, b) Image cryptée par l'algorithme TEA par blocs de 9 pixels dont 1 pixel en clair masqué, puis comprimée avec un $FQ = 80\%$, c) Image cryptée comprimée avec un $FQ = 100\%$, décomprimée, puis décryptée, d) Image cryptée comprimée avec un $FQ = 80\%$, décomprimée, puis décryptée, e) Image basse résolution reconstruite à partir des pixels clairs de l'image comprimée avec un $FQ = 100\%$, f) Image basse résolution reconstruite à partir des pixels clairs de l'image comprimée avec un $FQ = 80\%$.*

ramenés à un seul bloc qui contiendra :

- 4 octets de signalisation ($255 - 0 - 255 - 0$, par exemple),
- 2 octets contenant le nombre de blocs consécutifs K,
- 1 octet contenant l'indice de ce bloc particulier,
- 1 octet contenant le niveau de gris de ces blocs homogènes.

La signalisation représente une suite de 4 octets ayant des valeurs spéciales que l'on n'est pas censé rencontrer dans l'image. Si à la lecture de l'image cryptée on trouve ces 4 valeurs alignées, on sait qu'on est en présence d'un bloc spécial. Le second champ représente la longueur de la série sur 2 octets (maximum : 65536 blocs identiques consécutifs). L'indice est un compteur qui permet de différencier deux séries identiques. En effet, si par exemple il y a plusieurs séries de 2 blocs entièrement noirs dans l'image claire, sans ce compteur nous crypterions plusieurs fois le même bloc spécial, ce qui nous donnerait à nouveau dans l'image cryptée une dominance de certains niveaux de gris. Enfin, le dernier champ est le niveau de gris des huit pixels de chaque bloc homogène de la série.

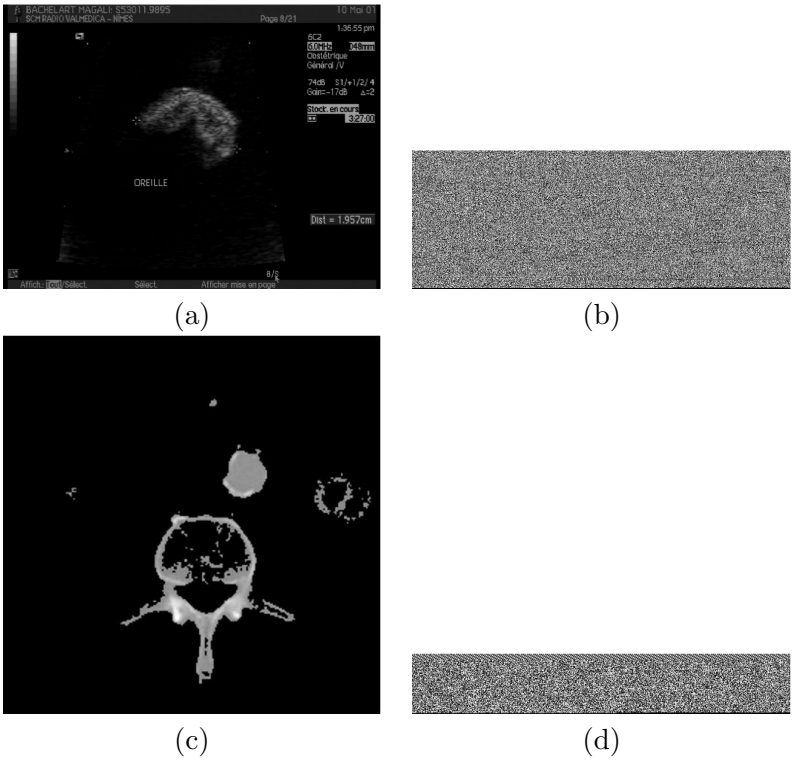
Ce bloc particulier se termine quand un nouveau bloc non homogène est rencontré dans l'image. Ce bloc particulier est alors chiffré de manière identique aux autres blocs. Notons que l'octet contenant l'indice de ce bloc particulier permet d'avoir des blocs chiffrés différents au cas où le nombre de blocs consécutifs homogènes serait trop souvent identique.

Dans le cadre de cette analyse, le nombre de blocs chiffrés ne peut être qu'inférieur ou égal au nombre de blocs de l'image initiale. Nous choisissons de garder une largeur fixe de l'image chiffrée et de réduire la hauteur de l'image en fonction du résultat de notre analyse.

4.6.2 Résultats de la méthode de crypto-compression

Nous appliquons dans cette section la méthode décrite section 4.6.1 à l'image médicale de la figure 83.a. Cette image contient de grandes zones de pixels noirs (= 0). La taille initiale de l'image est de 442384 octets (576 lignes $\times$ 768 colonnes), ce qui correspond à un chiffrement de 52298 blocs de 64 bits. Nous avons choisi d'utiliser l'algorithme DES. En fonction du regroupement des pixels pour créer les blocs de 64 bits nous obtenons un nombre variable de blocs noirs, décrit tableau19. Avec cette image médicale, nous avons analysé, tableau19, le nombre total de blocs noirs, la longueur maximale d'une série de blocs noirs, le nombre de séries ainsi que la longueur moyenne d'une série. Pour traiter cette image, nous avons choisi un regroupement des pixels en ligne (1×8) par rapport

au nombre de séries. Nous obtenons alors l'image crypto-comprimée de la figure 83.b. La taille de cette image n'est plus que de 215055 octets (280 lignes × 768 colonnes), soit un taux de compression égal à 2, sans aucune perte.



(a) (b)

(c) (d)

FIG. 83 – *Crypto-compression : a) Image médicale originale (442ko), b) Image (a) crypto-comprimée (215 ko), c) Image médicale originale (260ko), b) image (c) crypto-comprimée (40 ko).*

Dans le cas de l'image médicale de la figure 83.c, nous obtenons après crypto-compression une image 6.5 fois plus petite que l'originale, illustrée figure 83.d.

4.7 Conclusion et perspectives

Dans ce chapitre, nous avons montré comment les algorithmes classiques de chiffrement pouvaient être appliqué à des images. Les données images sont des données particulières du fait du fait de la taille des images et de l'information bidimensionnelle. Nous avons présenté un choix large d'algorithme par bloc ou par flot symétrique ou asymétrique. Nous en avons conclu que les algorithmes asymétrique tel que le RSA n'était pas adaptés aux images du fait de leur complexité du à l'utilisation de grands nombres premiers car une partie de la clef est connue (clef publique). Concernant les algorithmes symétriques,

forme des blocs	nbre blocs noirs	long. max série	nbre séries	longueur moyenne d'une série
1x8	29712	382	1220	24 blocs
2x4	32554	248	2010	16 blocs
4x2	33406	158	2842	11 blocs
8x1	32457	80	3883	8 blocs

TAB. 19 – *Nombre de blocs homogènes et longueur des séries en fonction du regroupement de pixels.*

les méthodes par bloc présentent des inconvénients quand l'image contient des zones homogènes. Dans le cas des algorithmes de chiffrement par flot, les zones homogènes ne sont plus visibles dans l'image cryptée. De plus les chiffrements par flot sont très rapides. Cependant, quelque soit l'algorithme de cryptage utilisé, il est alors difficile de comprimer l'image puisque théoriquement les redondances ont été supprimées durant la phase de cryptage et donc l'entropie devient maximale. De plus les algorithmes de chiffrement par bloc supportent très mal le bruit, en effet dès qu'un bit d'un bloc est altéré alors le bloc complet n'est décryptable. Dans le cas des chiffrements par flot, la robustesse au bruit semble plus importante. Dans ce chapitre nous avons présenté également une première approche de crypto-compression basée sur des images contenant des zones homogènes. Le premier objectif de cette méthode était de faire disparaître les zones homogènes, mais au final l'image est comprimée sans perte.

Les analyses développées dans ce chapitre sont à la base des méthodes de codage conjoint que nous présentons dans le chapitre suivant. L'idée restant devient alors non pas de cryptée une image avant la compression, mais de combiner les deux processus (compression et cryptage) en même temps.

Chapitre 5

Codages hybrides : cryptage, insertion de données cachées et compression

5.1 Introduction

Dans ce chapitre je développe des nouvelles méthodes de codage originales combinant toutes au moins deux types de codage différents, à savoir cryptage, insertion de données cachées et compression. Ces méthodes ont toutes pour objectif de protéger des données et sont issues des travaux présentés dans les chapitres 3 et 4.

La première méthode proposée, section 5.2, combine cryptage d'images et insertion de données cachées afin de rendre autonome un système de transmission sécurisé d'images. En effet, dans une approche classique à clef secrète, il faut utiliser un autre canal de transmission pour transférer la clef. A partir d'un algorithme de chiffrement par flot asynchrone robuste au bruit, nous proposons d'insérer dans l'image cryptée la clef secrète chiffrée par un algorithme asymétrique. Nous avons rappelé que les méthodes asymétriques ne conviennent pas aux images car trop longues en temps de calcul.

La seconde méthode, section 5.3, propose de combiner cryptage, compression et insertion de données cachées en créant un nouveau format d'image. Nous montrons dans cette méthode qu'en découpant l'image en deux parties (4 plans binaires de poids fort et 4 plans binaires de poids faible) il était possible dans la partie haute (plans binaires de poids forts) de l'image d'effectuer à la fois de l'insertion de données cachées et de la compression.

La troisième méthode, présentée section 5.4, propose de protéger la haute résolution d'une région d'intérêt de l'image fortement comprimée par JPEG. Actuellement, il est

possible avec JPEG2000 de ne pas comprimer une région d'intérêt de l'image tout en comprimant fortement le reste de l'image. Dans ce cas, à la décompression toute l'information est visible. Dans le cas de notre approche, la haute résolution n'est visible que si la personne qui décomprime l'image possède la clef secrète. En effet, nous évaluons la quantité de données perdues dues à la compression et nous insérons par données cachées ces pertes dans l'image comprimée.

Enfin, dans ce chapitre je propose, section 5.5, une méthode permettant de crypter de manière sélective les données de l'image tout en conservant un niveau de sécurité suffisant. Les avantages sont de pouvoir garder le taux de compression initial de l'image et de gagner en temps de calcul. En effet dans notre approche le cryptage des données est réellement effectué en même temps que la compression et ne rajoute aucune donnée supplémentaire.

Ces travaux ont été développés avec **S. Martineau**, **O. Léger**, **D. Falguère** et **A. Martin** dans le cadre de leur stage de DEA ainsi qu'avec **J. Rodrigues** dans le cadre de sa thèse. Cette partie a donné lieu aux publications suivantes : [Rodrigues 06, Puech 06, Amat 05, Toutant 05a, Puech 05, Puech 04a, Rodrigues 04a, Puech 04b].

5.2 Transfert autonome d'une image

5.2.1 Introduction

Nous avons vu précédemment que le transfert d'image augmente de plus en plus sur Internet et que la sécurité des transferts devenait très importante pour de nombreuses applications telles que la vidéo surveillance, les transmissions confidentielles et les applications médicales et militaires. Nous avons vu chapitres 3 et 4 qu'il y avait deux possibilités pour protéger des données durant leur transmission. La première possibilité consiste à crypter les images [Chung 98, Chang 01, Sinha 03]. Dans ce cas une clef est nécessaire pour décrypter l'image à la réception. La seconde possibilité consiste à utiliser des méthodes d'IDC qui ont pour objectif d'insérer le message à protéger dans une image. Ces deux approches sont complémentaires pour la protection. Dans cette section nous proposons une nouvelle méthode de protection combinant le cryptage d'images et l'IDC.

La protection de données qui dépend du secret de l'algorithme ne peut pas être considérée comme une bonne protection [Kerckhoffs 83, Schneier 95]. En effet les algorithmes d'IDC et de cryptage existants sont basés sur des clefs secrètes et non pas sur le secret des algorithmes. Dans l'approche traditionnelle, l'image est cryptée avec une méthode à clef secrète et la clef secrète est cryptée avec une méthode asymétrique à clefs publique et

privée. Le problème de cette approche est de transférer en même temps l'image cryptée et la clef secrète cryptée. Dans cette section nous allons développer un algorithme de chiffrement symétrique d'images pour un transfert sécurisé sans avoir besoin de transférer par un autre canal la clef secrète [Puech 04a, Puech 04b, Puech 06]. Pour cela, nous proposons de chiffrer la clef secrète avec un algorithme asymétrique et d'insérer cette clef cryptée dans l'image par IDC. Le fait d'insérer la clef dans l'image rend la méthode autonome et garantit l'intégrité des données. En effet, si l'image est attaquée durant le transfert (par modification de pixels ou découpage par exemple) alors le récepteur n'est plus capable d'extraire de l'image la clef cachée. Et par conséquent il n'est plus possible de décrypter l'image.

Nous avons vu chapitre 4 que les processus de cryptage pouvaient être symétriques ou asymétriques par bloc ou par flot. Nous avons vu que les algorithmes asymétriques ne sont pas appropriés au cryptage des images à cause de leur temps de calcul. Nous avons vu également que les algorithmes par bloc présentent trois inconvénients lorsqu'ils sont appliqués aux images. Le premier est quand nous avons des zones homogènes alors tous les blocs identiques sont cryptés de la même manière. Le second problème est que les algorithmes de chiffrement par bloc ne sont pas robustes au bruit. Le troisième problème concerne l'intégrité des données. Les figures 84 illustrent ce problème. A partir de l'image originale figure 84.a nous avons appliqué l'algorithme AES par bloc [AES01] avec une clef de 128 bits afin d'obtenir l'image cryptée figure 84.b. Si l'image cryptée est modifiée durant le transfert il n'est pas forcément possible de détecter la modification. Par exemple dans la figure 84.c nous avons permuté les quatre régions (modulo 128 bits) de l'image et dans la figure 84.d nous avons copié une petite région de l'image cryptée et nous avons collé cette région sur une autre zone de l'image. Après décryptage, il est possible de visualiser les images mais il n'est pas possible de garantir l'intégrité comme illustrée figures 84.e et f.

L'algorithme AES peut être utilisé avec quatre modes de cryptage[1] [AES01]: le mode ECB (Electronic CodeBook), les blocs sont chiffrés indépendamment les uns des autres et deux blocs clairs identiques (avec la même clef) fourniront deux cryptés identiques; le mode CBC (Cipher-Block Chaining), le chaînage fait que les blocs chiffrés dépendent de tous les blocs clairs précédents, une simple erreur binaire dans un bloc chiffré affecte le décodage de deux blocs; le mode CFB (Cipher FeedBack), le fait de changer l'ordre des blocs chiffrés affecte la phase de décryptage, une erreur binaire dans un bloc chiffré affecte le décodage

1. L'algorithme AES est détaillé dans la section 5.5.2.2.

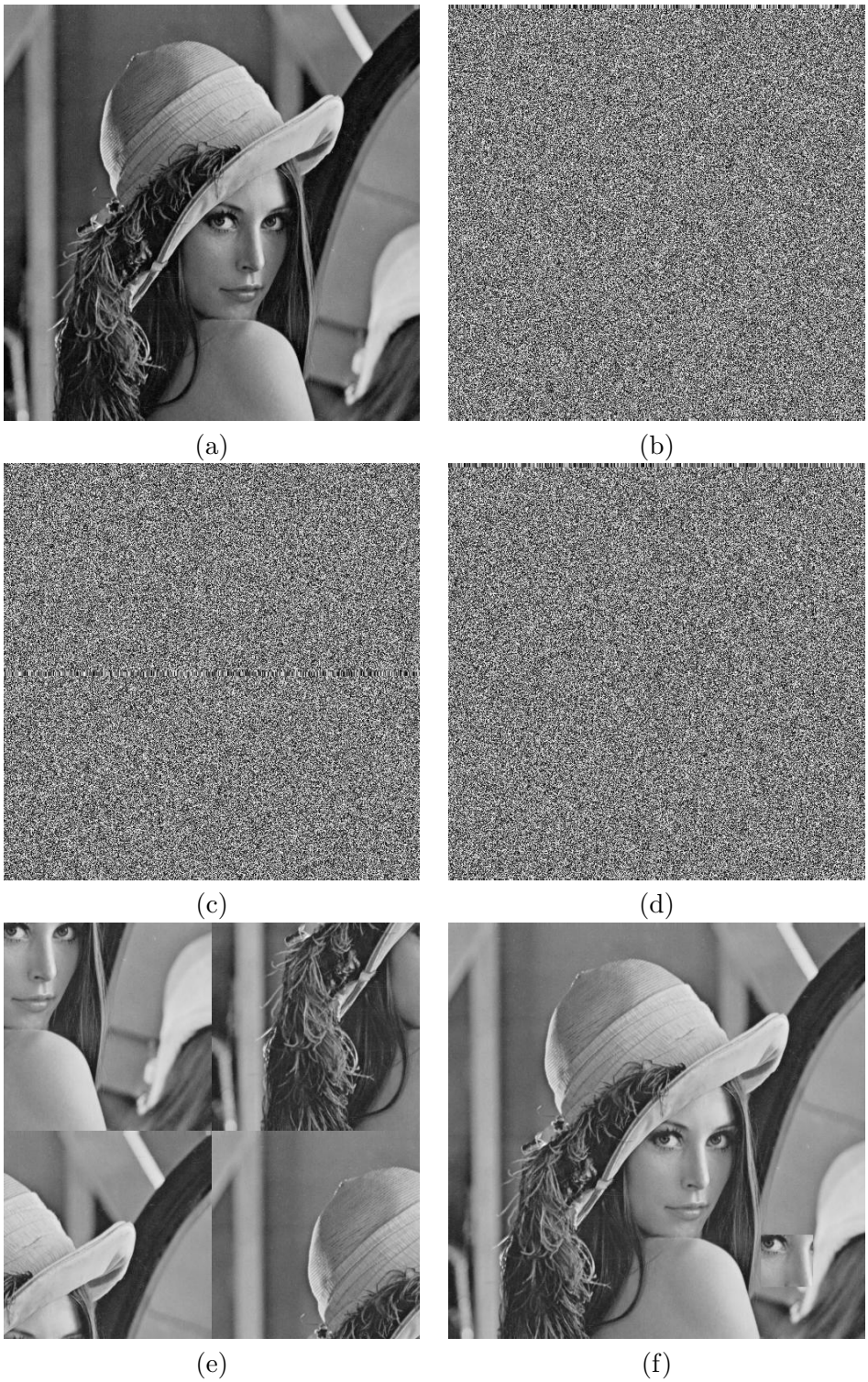


FIG. 84 – *a) Image originale de Lena, b) Image cryptée avec AES par bloc de 128 bits, c) Permutation de régions de l'image cryptée, d) Copie d'une région de l'image cryptée et collage sur une autre zone, e) Décryptage de (c), f) Décryptage de (d).*

de nombreux blocs chiffrés; le mode OFB (Output FeedBack), une erreur binaire affecte le décodage uniquement de ce caractère mais ne permet pas une synchronisation en cas de coupure d'une partie du message.

De manière générale les méthodes de chiffrement par flot sont plus robustes au bruit que les méthodes par bloc. Pour être robuste au bruit afin d'insérer un message dans l'image cryptée, nous avons choisi d'utiliser l'algorithme de chiffrement par flot asynchrone présenté section 4.3.3.

5.2.2 Analyse de la robustesse au bruit de la méthode de chiffrement par flot asynchrone

Dans cette section nous allons analyser la robustesse au bruit de la méthode de chiffrement par flot asynchrone présentée section 4.3.3. Avec un bruit additif n_i, chaque pixel crypté p'_i devient $\tilde{p}'_i$ tel que :

$$\tilde{p}'_i = p'_i + n_i, \tag{73}$$

avec n_i un bruit additif simulant le bruit dû à l'IDC, tel que $n_i \in \{-1,1\}$ et $Pr(-1) = Pr(1) = \frac{1}{2}$, où $Pr(x)$ est la probabilité d'avoir x.

Pendant le décryptage, à partir des équations (72) et (73) nous avons :

$$\tilde{p}_i = (\tilde{p}'_i - \sum_{j=1}^{k/2} \alpha_j \tilde{p}'_{i-j}) mod\ 256. \tag{74}$$

Supposons deux cas particuliers. Le premier cas est quand nous avons seulement un pixel bruité tous les $k/2$ pixels. Dans ce cas l'EQM' de l'image cryptée bruitée par rapport à l'image cryptée est :

$$\begin{aligned} EQM' &= \frac{1}{N} \sum_{i=0}^{N-1} (p'_i - \tilde{p}'_i)^2 \\ &= \frac{2}{k}. \end{aligned} \tag{75}$$

Par exemple, pour $k = 128$, l'$EQM' = 15.6\ 10^{-3}$, et le $PSNR = 66.19\ dB$. Dans ce premier cas, l'EQM de l'image décryptée est :

$$\begin{aligned} EQM &= \frac{1}{N} \sum_{i=0}^{N-1} (p_i - \tilde{p}_i)^2 \\ &= 1. \end{aligned} \tag{76}$$

La qualité de l'image décryptée ne dépend donc pas de la longueur k de la clef K, le $PSNR = 48.13\ dB$.

Le second cas particulier est quand tous les pixels cryptés sont bruités. Dans ce cas, l'EQM' de l'image cryptée est :

$$\begin{aligned} EQM' &= \frac{1}{N}\sum_{\imath=0}^{N-1}(p'_\imath - \tilde{p}'_\imath)^2 \\ &= \frac{1}{N}\sum_{\imath=0}^{N-1} n_\imath^2 \\ &= 1. \end{aligned} \tag{77}$$

Dans ce second cas la qualité de l'image cryptée ne dépend pas de la longueur k de la clef K, le $PSNR = 48.13\ dB$. Dans ce second cas particulier pour l'image décryptée, l'EQM est :

$$\begin{aligned} EQM &= \frac{1}{N}\sum_{\imath=0}^{N-1}(\sum_{\jmath=1}^{k/2}\alpha_\jmath n_\imath)^2 \\ &= 1 + \frac{3k}{8}. \end{aligned} \tag{78}$$

Cette valeur est obtenue à partir des équations (70) et (71) en considérant que nous avons $Pr(\alpha_\imath = 0) = \frac{1}{4}$, $Pr(\alpha_\imath = \pm 1) = \frac{1}{2}$ et $Pr_\alpha(\imath = \pm 2) = \frac{1}{4}$, où $Pr(x)$ est la probabilité d'avoir x. Dans ce second cas, avec $k = 128$ pour l'image décryptée le $PSNR = 31.23\ dB$.

En conclusion, dans le premier cas la différence de qualité entre l'image cryptée bruitée et l'image décryptée bruitée est de 18.6 dB. Dans le second cas, le bruit est plus intense, la différence de qualité diminue à 16.9 dB. Nous pouvons donc conclure que si il y a du bruit, quelque soit son intensité nous perdons plus de 16 dB de qualité. Mais même dans le second cas la qualité de l'image finale reste supérieure à 30 dB.

Afin de vérifier la robustesse au bruit de notre méthode, nous avons ajouté un bruit sur l'image cryptée, figure 74.e, avec un taux d'erreur binaire (TEB) de $1.95\ 10^{-3}$, figure 85.a. (ce TEB correspond à la modification du LSB de 1 pixel tous les 64 ($k/2$)) et avec un TEB de $1.55\ 10^{-1}$, figure 85.b (ce TEB correspond à la modification des LSB des tous les pixels). Les différences entre l'image cryptée et les deux images cryptées bruitées donnent un $PSNR = 66.15\ dB$ pour le TEB $1.95\ 10^{-3}$ (valeur théorique $PSNR = 66.19\ dB$) et un $PSNR = 48.13\ dB$ pour le TEB de $1.25\ 10^{-1}$ (valeur théorique $PSNR = 48.13\ dB$). Après décryptage des images cryptées bruitées, nous obtenons les images décryptées illustrées figures 85.c et d. Les différences entre l'image originale et les images décryptées bruitées, donne un $PSNR = 46.18\ dB$ (valeur théorique $PSNR = 48.13\ dB$) pour le TEB de $1.95\ 10^{-3}$ et un $PSNR = 27.74\ dB$ (valeur théorique $PSNR = 31.23\ dB$) pour le TEB de $1.25\ 10^{-1}$. Nous avons de petites différences entre les valeurs théoriques et notre exemple en partie à cause de l'équation (71) qui n'est pas exactement respectée.

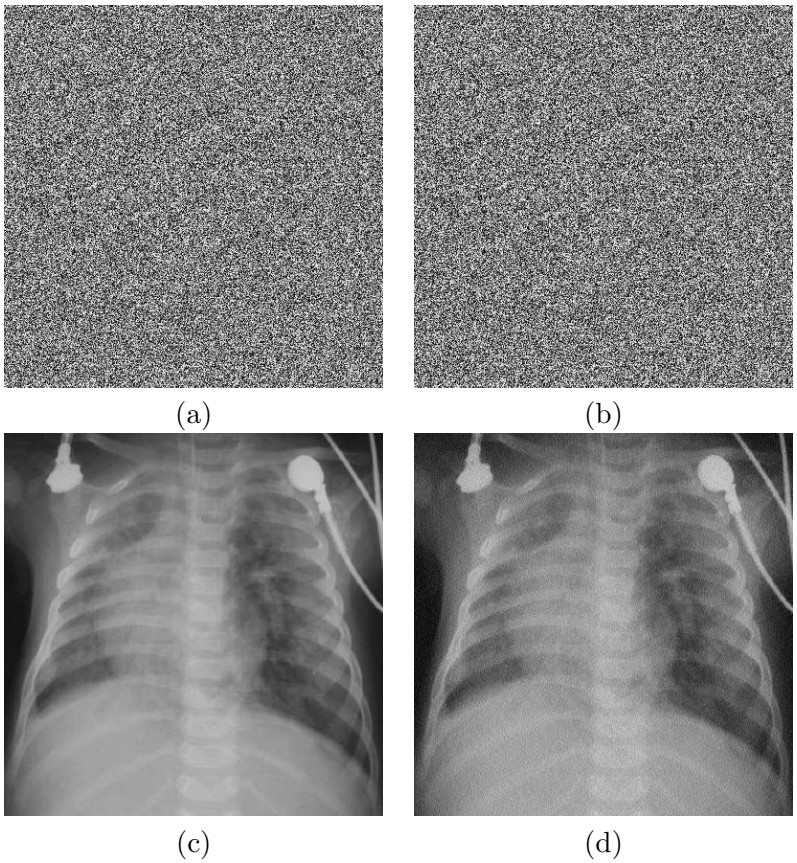


FIG. 85 – *a) Image cryptée avec l'algorithme de chiffrement par flot et bruitée (TEB=1.95 10^{-3}), b) Image cryptée avec l'algorithme de chiffrement par flot et bruitée (TEB=1.25 10^{-1}), c) Résultat du décryptage de l'image figure (a), d) Résultat du décryptage de l'image figure (b).*

Nous pouvons conclure de cette analyse qu'avec notre méthode de chiffrement par flot il était possible d'ajouter un bruit dans une image cryptée et d'obtenir un décryptage de l'image de bonne qualité. Dans la section suivante, le bruit cité sera l'insertion de la clef cryptée dans l'image.

5.2.3 Nouvelle méthode combinant cryptage et IDC

Dans cette section nous proposons une nouvelle méthode combinant l'algorithme de chiffrement par flot présenté chapitre 4 avec une clef secrète pour l'image et un algorithme asymétrique pour chiffrer la clef secrète. Ensuite nous utilisons la méthode d'IDC, présentée chapitre 3, pour insérer la clef cryptée dans l'image cryptée. D'un point de vue pratique, si une personne A envoie par réseau une image à B, l'émetteur A utilisera l'algorithme de chiffrement par flot avec la clef secrète K pour crypter l'image. Ensuite le problème est pour transmettre la clef K. Afin de transmettre cette clef A peut chiffrer la clef K en utilisant un algorithme à clef publique tel que RSA par exemple [Sec03]. Soit pub (e,n) la clef publique et $priv$ (d,n) la clef privée pour RSA avec $e = d^{-1}\%n$, alors A a ses clefs publique et privée pub_a (e_a,n_a) et $priv_a$ (d_a,n_a), et B ses clefs publique et privée pub_b (e_b,n_b) et $priv_b$ (d_b,n_b).

Par conséquent A génère une clef secrète K pour cette session et chiffre l'image avec l'algorithme de chiffrement par flot. Ensuite A chiffre la clef K avec l'algorithme RSA en utilisant sa clef privée $priv_a$ afin d'obtenir un clef cryptée K' telle que :

$$K' = K^{d_a}\ mod(n_a). \tag{79}$$

Cette clef cryptée K' est cryptée une seconde fois avec RSA en utilisant la clef publique pub_b de son correspondant B afin de générer K'':

$$K'' = K'^{e_b}\ mod(n_b). \tag{80}$$

Pour les équations (79) et (80), si le message à chiffrer est plus long que le modulo alors le message doit être coupé en partie plus petite que n_a ou n_b.

Dans notre méthode de combinaison la taille du message à insérer dans l'image dépend de la taille de la clef publique du récepteur. Cette taille est connue par l'émetteur A et le récepteur$B$. Nous pouvons donc calculer le facteur d'insertion et calculer le nombre de blocs nécessaires pour la méthode d'IDC. Cette clef K'' est donc insérée dans l'image chiffrée en utilisant la méthode d'IDC présentée chapitre 3. Finalement, A envoie l'image à B comme présentée figure 86. Cette procédure de cryptage K avec $priv_a$ et pub_b assure

l'authenticité et seul B peut décrypter l'image envoyée. Le fait d'insérer la clef dans l'image rend la méthode autonome et garantit l'intégrité. En effet, si durant le transfert l'image est attaquée alors il n'est plus possible à la réception d'extraire la bonne clef et donc de décrypter l'image.

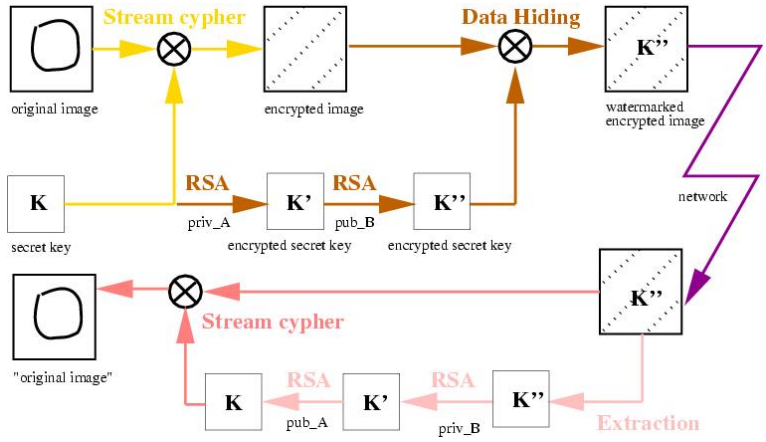


FIG. 86 – *Combinaison d'un cryptage à clef secrète, d'un cryptage à clef publique et d'une méthode d'IDC.*

La personne B reçoit l'image cryptée et marquée et peut alors extraire la clef cryptée K''. Il peut alors authentifier A et décrypter la clef K'' en utilisant sa clef privée $priv_b$ et la clef publique pub_a de A telles que :

$$K = (K''^{d_b} \% n_b)^{e_a} \ mod(n_a). \tag{81}$$

Avec la clef obtenue K, B peut déchiffrer l'image et la visualiser. Si B veut envoyer une nouvelle image à A, il doit générer une nouvelle clef secrète K_1 pour cette nouvelle session. Le processus sera le même mais les clefs publiques et privée pour RSA ne seront pas appliquées dans le même ordre. Même si cinq clefs sont nécessaires pour chaque session, celles-ci sont transparentes pour les utilisateurs. En effet les clefs privées sont associées au logiciels utilisés et pour les deux correspondants il n'est pas nécessaire de connaître la valeur de la clef secrète qui est cryptée et insérée dans l'image. Toutefois, pour chaque session la valeur de la clef secrète K doit changer. Sinon, si la clef était toujours la même tous les gens qui ont le logiciels pourraient décrypter les images.

5.2.4 Résultats

Dans cette section nous présentons les résultats de la méthode de combinaison. A partir de l'image originale de Lena (512×512 pixels), figure 87.a, nous avons appliqué

notre méthode de chiffrement par flot avec une clef K de 128 bits, afin d'obtenir l'image cryptée figure 87.b. Si nous décryptons cette image, nous pouvons noter qu'il n'y a aucune différence entre celle-ci et l'originale. Avec $K = F8DCBA98C6543210FEDCBA987F5432B0$ nous obtenons les valeurs de $\alpha(\imath)$ et $p(\imath)$, montrées tableau 20.

TAB. 20 – *Valeurs de $\alpha(\imath)$ et $p(-\imath)$.*

$\alpha(\imath)$	2 -2 1 -1 -2 0 2 -1 1 2 1 1 1 0 1 -1 -2 -1 0 1 0 0 0 -1 -1 -2 -1 1 -1 0 -1 -1 2 2 -2 1 2 0 -2 -1 1 2 1 1 1 0 1 -1 0 -2 -2 2 0 0 0 -1 -1 -2 -1 1 1 2 -1 -1
$p(-\imath)$	248 241 227 198 141 27 55 110 220 185 114 229 203 151 46 93 186 117 234 212 169 83 166 76 152 49 99 198 140 24 49 99 198 140 25 50 101 202 149 42 84 168 80 161 67 134 12 25 50 100 200 144 33 66 132 8 16 33 67 135 15 31 63 127

Nous avons crypté la clef K de 128 bits deux fois avec l'algorithme RSA afin d'obtenir K''. Du fait de la longueur de la clef publique de B, la longueur de K'' est proche de 512 bits. Ensuite avec la méthode d'IDC basée sur la DCT nous avons insérer la clef K'' dans l'image cryptée, figure 87.c. Le facteur d'insertion est de 1 bit tous les 8 blocs de 64 pixels. La différence entre l'image cryptée et l'image cryptée marquée est présentée figure 87.d. Les blocs utilisés pour l'IDC sont visibles, le $PSNR = 39.25\ dB$.

Finalement, après décryptage de l'image cryptée et marquée figure 87.c nous obtenons l'image finale illustrée figure 87.e. La différence entre l'image originale et l'image finale est présentée 87.f. Nous voyons dans cette figure que les différences entre les deux images ($PSNR = 38.75dB$) ont été diffusées dans toute l'image. Cependant, du fait que la valeur moyenne des coefficients $\alpha(\imath)$ est égale à zéro le bruit dû à l'IDC est atténué durant la phase de décryptage.

Afin de comparer notre résultat nous avons appliqué notre méthode d'IDC sur l'image Lena cryptée en utilisant l'algorithme AES avec le mode ECB, Figure 88.a, et avec le mode OFB, Figure 88.b, qui sont deux modes de chiffrement par flot. Après décryptage de l'image marquée et chiffrée par AES en mode ECB nous obtenons l'image illustrée figure 88.c. L'image différence entre l'image originale et l'image décryptée, figure 88.e, montre que les variations sont très importantes, le $PSNR = 14.81dB$. Nous pouvons remarquer que la largeur des blocs faux est égale à 128 bits (16 pixels). Après décryptage de l'image marquée et chiffrée par AES en mode OFB nous obtenons l'image illustrée figure 88.d. La différence entre les images originale et décryptée, figure 88.e, montre que les variations ne sont pas diffusées par ce mode. La qualité de l'image est bonne, le $PSNR = 37.23dB$.

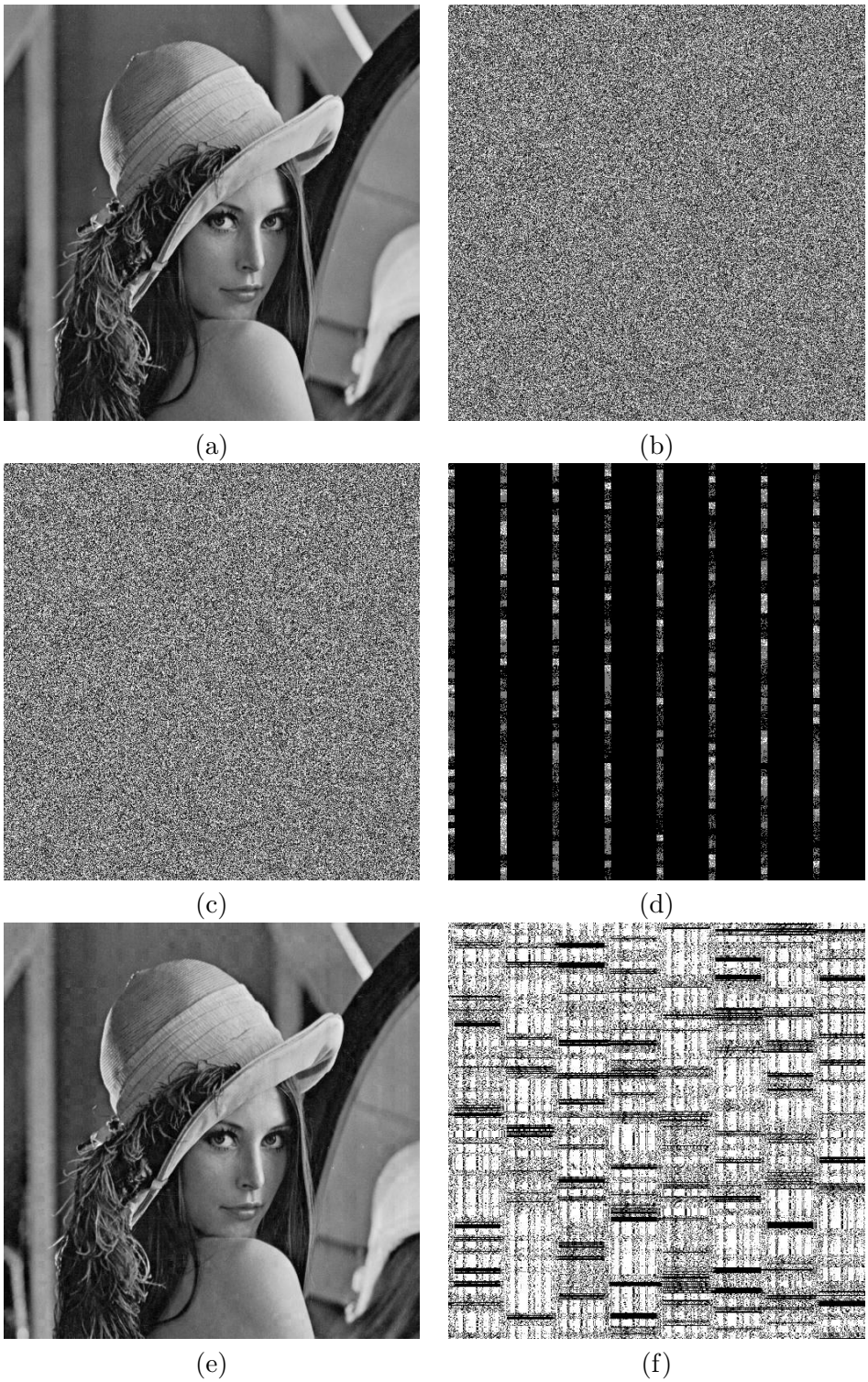


FIG. 87 – *a) Image originale, b) Image cryptée par flot avec une clef de 128 bits, c) Image (b) marquée avec la clef secrète cryptée, d) Différence entre les images (b) et (c), e) Décryptage de l'image (c), f) Différence entre l'image originale image (a) et (e).*

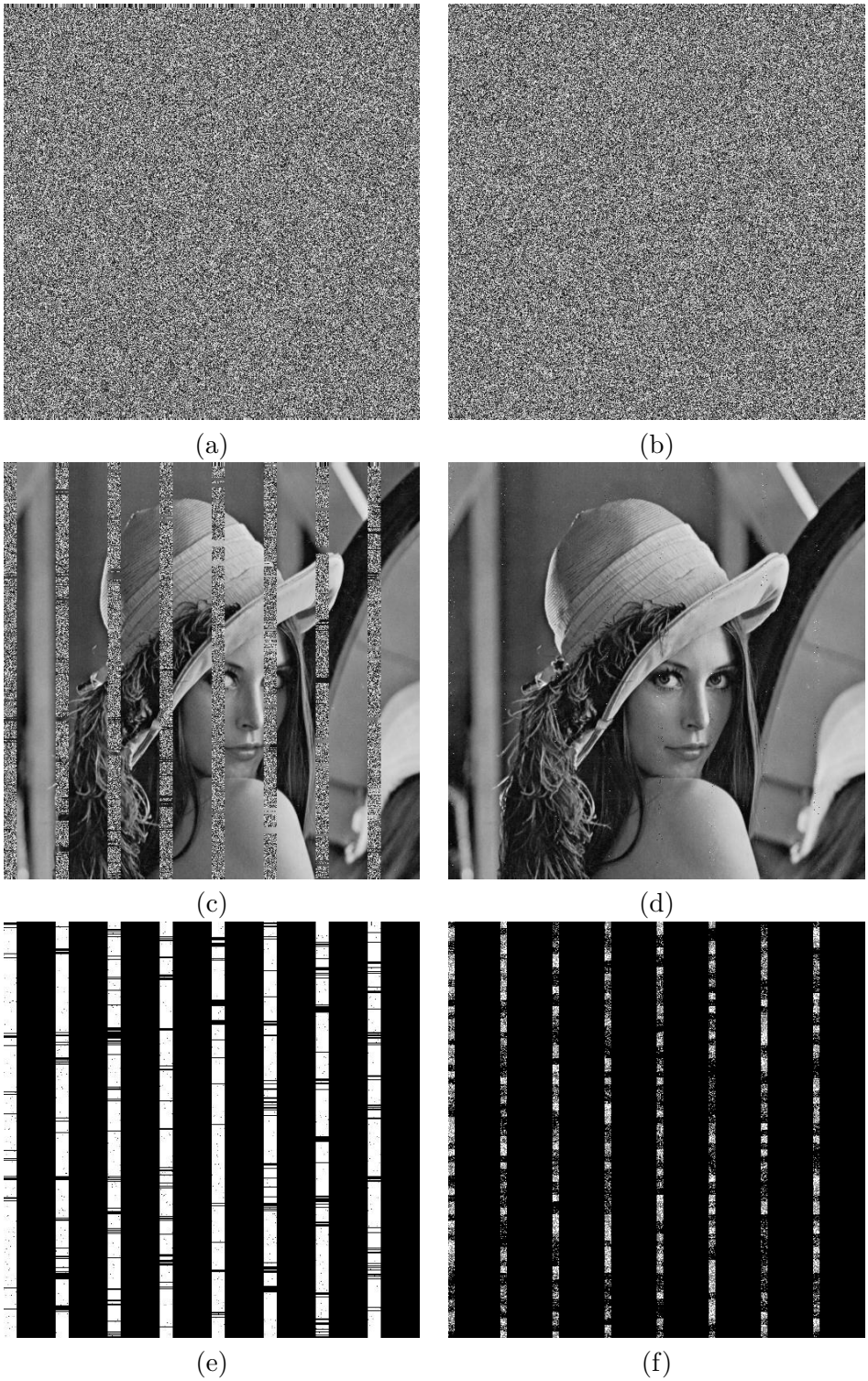


FIG. 88 – *a) Image Lena chiffrée avec l'algorithme AES en mode ECB et marquée, b) Image Lena chiffrée avec l'algorithme AES en mode OFB et marquée, c) Résultat du décryptage de (a), d) Résultat du décryptage de (b). e) Différence entre l'image originale et (c), f) Différence entre l'image originale (d).*

Avec notre méthode nous gagnons entre 1 et 2 dB au niveau du PSNR. Le problème avec le mode OFB d'AES est le débordement. Quelques pixels noirs deviennent blancs et des pixels noirs deviennent blancs.

En conclusion avec notre méthode combinant cryptage et IDC il est possible d'avoir un système de transmission autonome et de garantir l'intégrité des données. Notre méthode de chiffrement asynchrone est robuste au bruit et par conséquent nous pouvons insérer un message dans une image cryptée sans perturber la phase de décryptage. Nous avons comparé notre méthode de chiffrement par flot aux modes ECB et OFB d'AES mais nous n'obtenons pas le même niveau de qualité. De plus, avec notre méthode d'IDC, si il y a du bruit durant le transfert nous sommes toujours capables d'extraire le bon message et donc de décrypter la bonne clef.

5.2.5 Conclusion

Dans cette section, nous avons présenté une méthode qui combine cryptage et IDC. Il est ainsi possible d'avoir un système de transmission autonome permettant de garantir l'intégrité des données. Nous avons utilisé à la fois les avantages de cryptages symétrique et asymétrique. Dans la méthode proposée, nous avons choisi de chiffrer avec notre méthode de chiffrement par flot asynchrone et de chiffrer la clef secrète avec un algorithme asymétrique.

Nous avons vu que notre méthode était robuste à une certaine quantité de bruit. Pour insérer la clef secrète cryptée nous avons utilisé notre méthode d'IDC basée sur la DCT. De ce fait cette méthode d'IDC est robuste au bruit pouvant survenir durant le transfert.

Nous avons analysé la robustesse au bruit de notre méthode de cryptage appliquée sur une image médicale. Finalement nous avons présenté un résultat complet de la méthode et avons comparé dans la combinaison notre algorithme de chiffrement aux modes de chiffrement par flot de l'algorithme AES. Notre méthode de chiffrement permet de conserver une meilleure qualité pour l'image finale.

Nous avons vu dans le chapitre précédent que les attaques statistiques n'étaient pas possible du fait d'une entropie élevée. Mais en perspective de cette méthode de combinaison nous pensons évaluer d'autres types d'attaques afin de garantir un meilleur niveau de sécurité.

5.3 Crypto-compression réversible

5.3.1 Introduction

Dans cette section nous présentons une méthode de protection de données combinant IDC sans perte, compression et cryptage en créant un nouveau format de données. La plupart des méthodes d'IDC sont des méthodes dites irréversibles ou avec pertes [Fridrich 98]. Cela signifie que après l'IDC l'image support a été modifiée. Nous ne pouvons donc pas retrouver l'image originale et certaines informations importantes peuvent être perdues. Pour des applications particulières telles que l'imagerie médicale, l'image originale doit absolument être conservée. Les méthodes d'IDC réversibles sont la solution pour ce type de problème. Ces méthodes peuvent être utilisées pour associer à l'image des informations cruciales sans changer le contenu de l'image. Plusieurs méthodes ont été développées dans ce sens [Honsinger 01, Fridrich 02b], et avec conservation de la taille initiale de l'image [Fridrich 04]. Il existe également de nombreuses méthodes de compression d'images sans perte [Pennebaker 93, Wu 96, Maniccam 01]. Toutefois, aucune méthode propose à la fois une compression, un chiffrement et une IDC sans perte. En fait la plupart des méthodes d'IDC réversibles augmente la taille de l'image originale. Il nous a donc semblé nécessaire de trouver des nouvelles méthodes permettant d'insérer une grande quantité d'information dans l'image sans augmenter la taille de celle-ci et sans en modifier le contenu tout en la chiffrant. Dans cette section, nous développons dons une méthode réversible combinant cryptage et IDC tout en diminuant la taille de l'image [Rodrigues 04a, Rodrigues 06].

5.3.2 Méthode proposée

La méthode proposée est résumée figure 89. Premièrement l'image originale est décomposée en deux demi-images (SPI pour semi-pixel image). Ensuite la SPI contenant les plans de poids forts est comprimée et marquée avec un message binaire. Cette nouvelle image est alors mélangée avec la seconde SPI pour être chiffrée avec un algorithme à clef secrète.

La caractéristique la plus importante de la méthode proposée est l'utilisation d'une seule procédure pour réaliser l'ensemble des traitements. Cette approche diminue donc le temps de traitement et peut être utilisée dans des environnements de faible puissance tels que les téléphones portables. De plus, il ne nous semble pas raisonnable d'utiliser trois étapes séparées avec des algorithmes standards. Par exemple, si nous appliquons le RS4RLE [Fridrich 04] pour une IDC réversible, il n'est alors plus possible de comprimer l'image avec un algorithme de compression sans perte tel que JPEG2000, RLE or LZW.

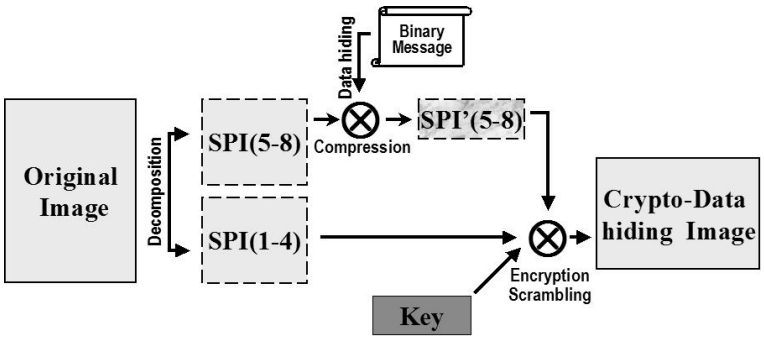


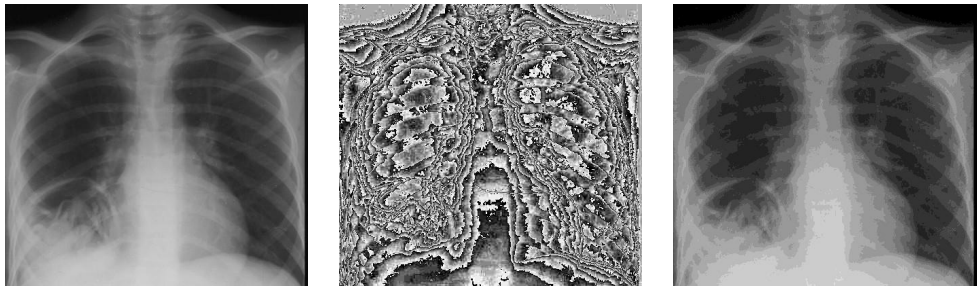
FIG. 89 – *Présentation générale de la méthode.*

En effet, la méthode RS4RLE compacte déjà le contenu de l'image afin d'y insérer des données. Par conséquent un algorithme de compression sans perte n'améliorera pas le taux de compression après la méthode RS4RLE. De plus l'application de trois procédures séparées augmente le temps de traitement.

5.3.2.1 Décomposition de l'image

La méthode décompose à la base l'image originale en deux SPIs (semi-pixel images). La première SPI, nommée SPI(1-4), est composée des quatre LSBs de l'image originale. La seconde SPI, nommée SPI(5-8), est composée des quatre autres bits restant comme présenté figure 89.

Cette idée est basée sur les travaux de [Maniccam 01], qui montre que le facteur de compression est plus élevé dans les quatre MSB. Nous remarquons, figures 90.a, b et c, que la partie la plus représentative de l'image est dans la SPI(5-8). De plus dans la SPI(5-8) il y a un grand nombre de zones homogènes. Nous allons donc appliquer un algorithme de compression sur la SPI(5-8).



(a) (b) (c)

FIG. 90 – *a) Image originale, b) SPI(1-4) : demi-image contenant les LSBs, c) SPI(5-8) : demi-image contenant les MSBs.*

5.3.2.2 Compression de SPI(5-8)

La SPI'(5-8) est obtenue à partir de la compression de la SPI(5-8) avec une variante du RLE (Run Length Encoding). RLE est un algorithme de compression sans perte qui affecte des codes courts à des longues séquences de symboles identiques. Celui-ci est utilisé pour des images contenant des zones homogènes. Quand l'algorithme RLE détecte une longue séquence homogène, il représente cette séquence par un bloc spécial contenant un drapeau de signalisation , la longueur de la séquence ainsi que la couleur des pixels composant cette séquence. Pour notre méthode nous proposons une variation du RLE en utilisant deux sortes de blocs spéciaux. Ces blocs sont fonction du contenu de l'image :

- Bloc de 8 bits (B_8), avec zéro ou un bit pour l'IDC :
  - 1 bit ($valeur = 0$) pour identifier le type de bloc.
  - H_8 bit pour l'IDC, avec $0 \leq H_8 \leq 1$.
  - $3 - H_8$ bits pour la longueur de la séquence.
  - 4 bits pour la couleur.
- Bloc de 16 bits (B_{16})
  - 1 bit ($valeur = 1$) pour identifier le type de bloc.
  - H_{16} bits pour l'IDC, avec $2 \leq H_{16} \leq 7$.
  - $11 - H_{16}$ bits pour la longueur de la séquence.
  - 4 bits pour la couleur.

Le nombre de bits disponibles pour l'IDC dans les blocs H_8 et H_{16} dépend respectivement de la longueur moyenne des segments l_8 et l_{16} :

$$\left\{ \begin{array}{lcl} H_8 & = & \left\{ \begin{array}{ll} 1 & \text{si} \quad l_8 < 4 \\ 3 - \lceil log_2(l_8) \rceil & \text{si} \quad 4 \leq l_8 \leq 8 \end{array} \right. \\ H_{16} & = & 11 - \lceil log_2(l_{16}) \rceil \end{array} \right. , \tag{82}$$

où $8 < l_{16} \leq 512$.
Les blocs B_8 peuvent donc avoir zéro ou un bit pour l'IDC. Les blocs B_{16} ont au moins deux bits fixes pour l'IDC. Cependant, le nombre total de bits utilisés pour l'IDC dépend de la longueur des séquences car certains bits du bloc peuvent être utilisés pour l'IDC, figure 91. Par exemple, si la longueur du segment des blocs B_{16} $l_{16} = 60$, alors $H_{16} = 5$ bits par bloc pour l'IDC.

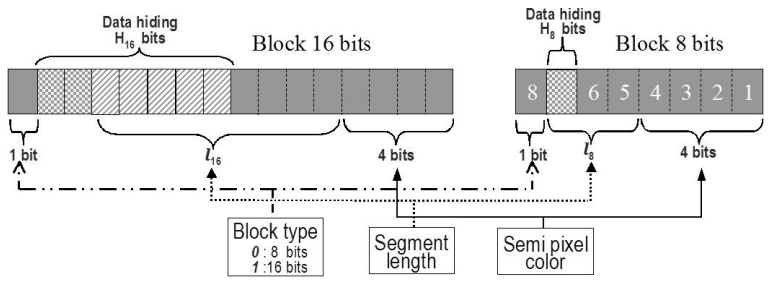


FIG. 91 – *Contenu des blocs B_{16} et B_8.*

Dans les SPI(5-8), la quantité de données cachées W est :

$$W = n_8 \times H_8 + n_{16} \times H_{16}, \tag{83}$$

où n_8 et n_{16} sont respectivement le nombre de blocs B_8 et B_{16}.

Comme la compression cherche des séquences homogènes, le chemin utilisé pour parcourir la SPI est important. L'objectif principal de cette étude n'est pas de trouver le meilleur parcours afin d'obtenir le plus grand taux de compression, mais de trouver rapidement un parcours permettant d'associer compression et IDC. Cependant, il existe de nombreuses possibilités pour lire l'image [Maniccam 04], pour notre méthode la SPI(5-8) est parcourue suivant seulement trois parcours comme illustrés figure 92.

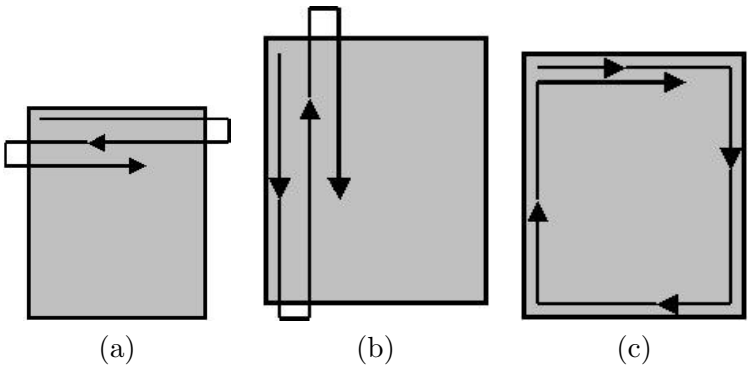


FIG. 92 – *Trois différents parcours pour la SPI(5-8) a) ligne, b) colonne, c) spirale.*

Le contenu de l'image défini le nombre et le type de blocs. Si SPI(5-8) a une remarquable somme de régions homogènes et que la longueur de ces régions est très longue alors l'algorithme utilisera beaucoup de blocs B_{16}. Si ses régions homogènes sont petites l'algorithme choisira des blocs B_8.

L'OLLS (Optimal Length of the Longest Sequence) est déterminé dynamiquement pour chaque type de bloc et en fonction de la taille finale de l'image et de la taille du message à insérer. L'algorithme parcourt l'image en considérant que la plus longue séquence l_8 pour les blocs de type 8 peut-être {4 codable sur 2 bits ou 8 codable sur 3 bits}. Le même traitement est utilisé pour les blocs de type B_{16} en utilisant l'ensemble de valeurs {16, 32, 64, 128, 256 et 512}. Finalement l'algorithme propose l'OLLS ayant un message le plus grand possible avec une taille d'image la plus petite possible.

5.3.2.3 IDC réversible

Comme l'algorithme RLE a déjà été appliqué nous avons maintenant de la place pour insérer le message. Le processus d'insertion est illustré figure 93.

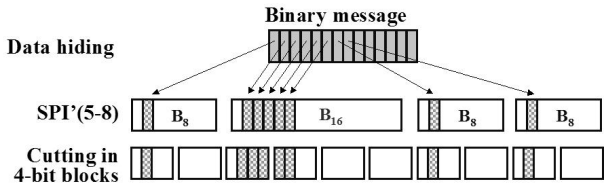


FIG. 93 – *Schéma d'IDC et découpage de SPI'(5-8).*

L'insertion des données est effectuée bit par bit. Le données sont dispersées dans toute la SPI(5-8) en fonction de la disposition des blocs B_8 et B_{16} dans l'image. Après l'IDC, la SPI(5-8) est découpée en portion de quatre bits, figure 93, afin de préparer le phase de cryptage.

5.3.2.4 Procédure de chiffrement

La phase de chiffrement se décompose en trois parties qui sont le mélange (scrambling) la fusion et le cryptage. Le mélange et la fusion sont dus au fait de la décomposition originale pour la phase de compression et d'IDC. Le fait que l'IDC diffuse l'information dans toute la SPI(5-8) ne peut pas être considéré comme une méthode de cryptage [Kerckhoffs 83]. En fait la méthode proposée est basée sur des opérations de l'algorithme AES. Notre méthode s'appuie sur des clefs de longueur 128 bits[2]. La clef a un rôle important dans la phase de cryptage. Elle est divisée en sous-clefs de 8 bits chacune. Ces sous-clefs sont alors XORées avec les pixels obtenus de la fusion et du mélange de SPI'(5-8)

2. Une clef de longueur au moins égale à 80 bits est nécessaire pour se protéger contre des attaques brutales.

avec SPI(1-4). Le clef est également utilisée comme générateur de nombre pseudo-aléatoire (GNPA). Ce GNPA produit de nombres utilisés dans trois partie du traitement :

- Pour indiquer, dans l'image finale, la position des données critiques pour l'extraction.
- Pour déterminer la SPI(1-4).
- Pour déterminer le type d'inversion :
  - Pair : (5-8) (1-4).
  - Impair : (1-4) (5-8).

Le premier nombre aléatoire produit est utilisé seulement pour l'opération d'extraction. Ensuite si le processus de chiffrement génère un nombre pair alors la fusion se fait dans l'ordre SPI(5-8) et SPI(1-4), sinon dans l'ordre inverse. L'algorithme AES mélange également les données (ShiftRows(), MixColumns()) afin d'interdire une extraction illégale [AES01]. L'algorithme de chiffrement propose alors une partie de la clef pour appliquer une opération XOR avec le pixel créé par fusion comme illustré figure 94. Comme la taille de SPI'(5-8) est plus petite que SPI(1-4) ce processus est appliqué jusqu'à la fin de SPI'(5-8). Les données restantes de SPI(1-4) sont fusionnées entre elles de manière aléatoire puis chiffrées de la même manière.

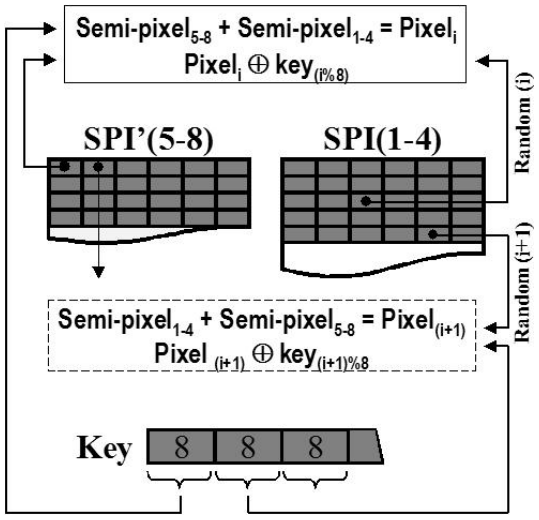


FIG. 94 – *Algorithme de chiffrement.*

L'étape finale de l'algorithme de chiffrement est l'insertion des données critiques pour l'algorithme d'extraction. Ces données sont insérées dans huit octets. Le premier nombre généré par un GNPA (entre 0 et la moitié de la taille de l'image originale) est utilisé pour

insérer les données dans l'image résultante. Ces huit octets sont composés de :

- 1 bit pour préciser H_8,
- 3 bits pour H_{16},
- 2 bits pour le mode de lecture (ligne, colonne ou spirale),
- 12 bits pour le nombre de colonnes de l'image originale,
- 12 bits pour le nombre de lignes de l'image originale,
- 16 bits pour la taille des données cachées,
- 18 bits pour la taille de SPI'(5-8).

5.3.2.5 Extraction

La phase d'extraction utilise la clef secrète comme semence pour le GNPA. Le premier nombre indique la position de l'information nécessaire pour l'extraction. Ensuite, avec la taille de SPI'(5-8) et de SPI(1-4) (moitié de l'image originale) et la clef secrète nous pouvons recréer SPI(1-4) et SPI'(5-8). Avec H_8, H_{16} et la taille du message, nous pouvons extraire l'information cachée. Finalement, nous pouvons reconstruire l'image originale sans perte.

5.3.3 Résultats expérimentaux

Image	KHVomique		Lena	
Taille image originale (octets)	185090		262144	
Optimale	Taille W	Compression	Taille W	Compression
Type de parcours	ligne	spirale	ligne	colonne
H_8 | H_{16}	1 | 7	0 | 3	1 | 7	0 | 4
n_8	11301	12467	74884	65881
n_{16}	14598	6998	14885	6140
Taille image finale (octets)	133354	119082	236032	209408
Compression τ	1.39	1.55	1.11	1.25
W (octets)	14497	2698	22690	3245
W pourcentage	7.83 %	1.46 %	8.66 %	1.24 %
Entropie (bits/pixel)	7.96	7.91	7.99	7.99

TAB. 21 – *Résumé des résultats.*

La méthode proposée a été testée sur plus de vingt images en niveaux de gris. Toutefois nous présentons les résultats tableau 21, pour les deux images illustrées figures 95.a et 96.a. Elles ont été chiffrées avec une clef de 128 bits et complètement remplies avec un

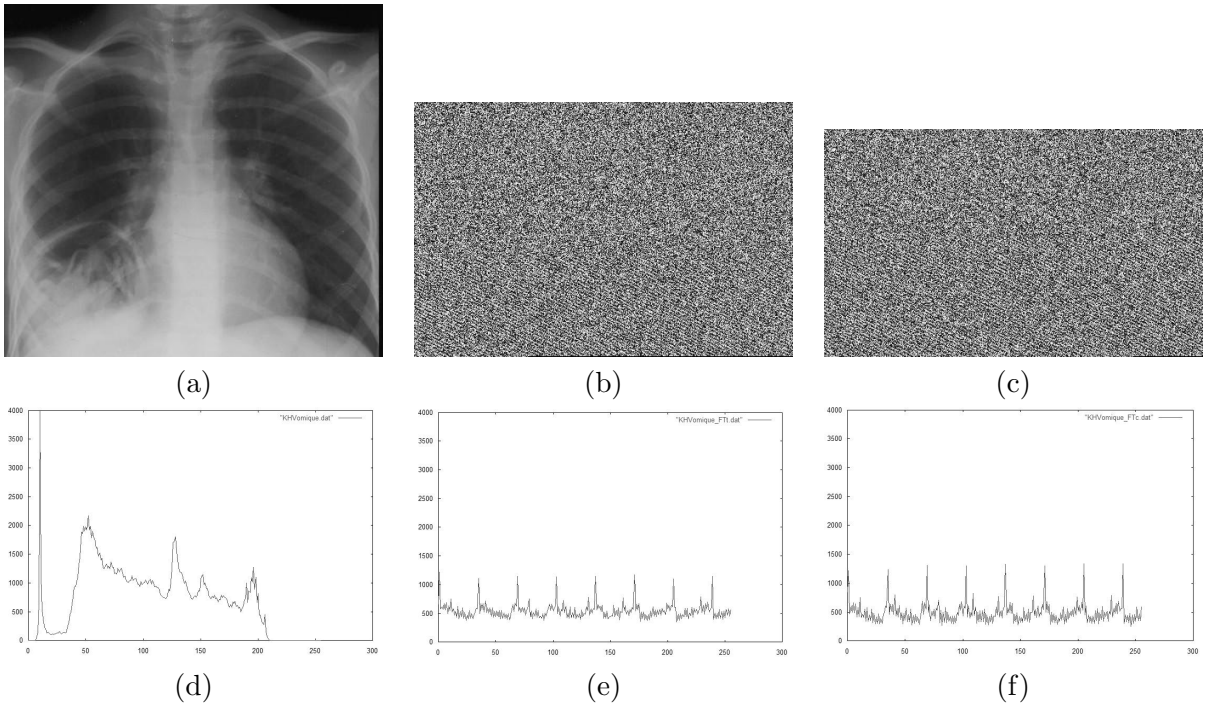


FIG. 95 – *a) Image originale KHVomique, b) Image cryptée et marquée avec une insertion optimale, c) Image cryptée et marquée avec une compression optimale, d) Histogramme de l'image originale, e) Histogramme de l'image (b), f) Histogramme de l'image (c).*

message binaire. La figure 95.b montre l'image cryptée et marquée avec un espace optimal pour l'IDC. Nous avons pu insérer un message de 14497 octets (compte rendu complet de l'examen d'un patient). Même en insérant cette importante quantité d'information nous avons atteint une image finale de taille 27.95% plus petite que l'originale. L'image finale a pour taille 133354 octets. La figure 95.c présente l'image chiffrée et marquée avec une compression optimale. Nous avons inséré un message binaire de 2698 octets (résumé médical du patient) et nous avons obtenu une taille finale de 119082 octets. Ceci correspond à un taux de compression de 1.55 soit une image finale 35.66% plus petite que l'originale. Nous montrons également les histogrammes des images citées : l'original, figure 95.d, et ceux des images crypto-marquées figures 95.e et 95.f. Ces deux derniers histogrammes présentent une très grande entropie proche de 8 bits/pixel et confirme un niveau de sécurité du chiffrement. Pour l'image illustrée figure 95.b, la séquence de l'OLLS pour les blocs B_8 est 4 et pour les blocs B_{16} est 16. Pour l'image figure 95.c, ces valeurs sont respectivement 8 et 256. Dans le tableau 21 nous pouvons noter que la capacité d'IDC de la méthode proposée dépend fortement des caractéristiques de l'image originale.

La même analyse peut être effectuée pour l'image de Lena figures 96. La taille optimale

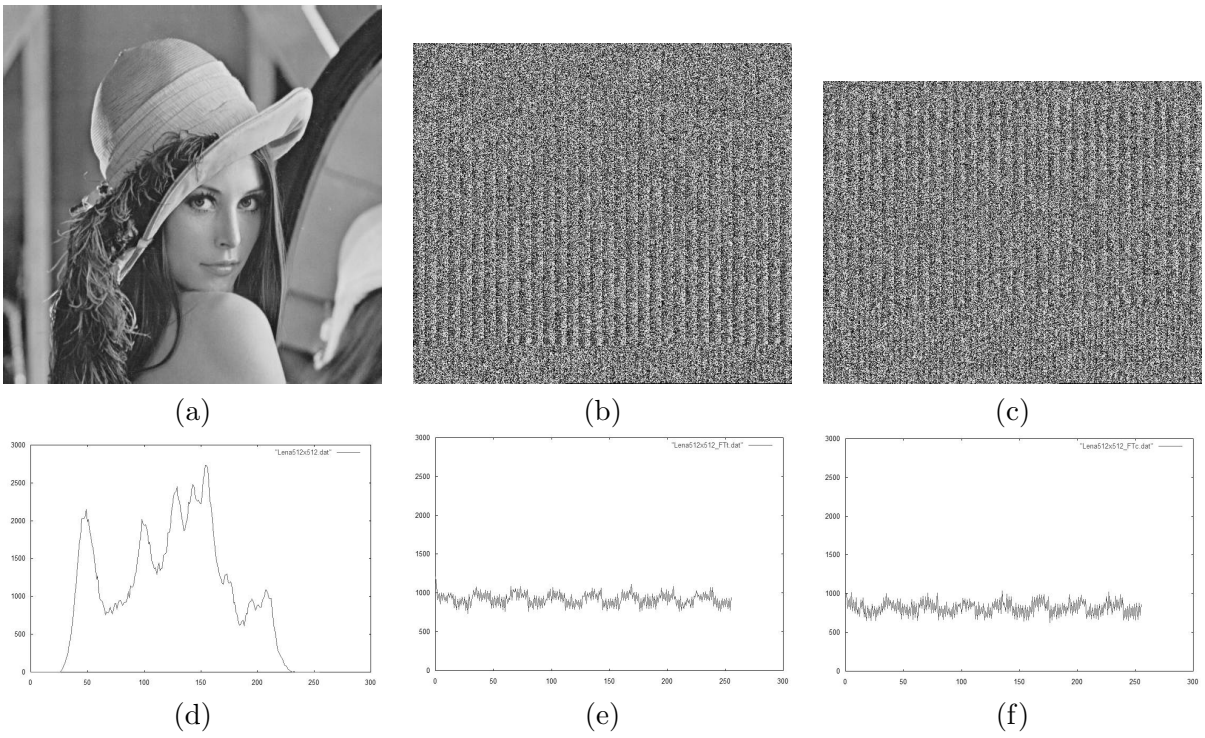


FIG. 96 – *a) Image originale Lena* 512×512*, b) Image chiffrée et marquée avec une place optimale pour le message, c) Image chiffrée et marquée avec une compression optimale, d) Histogramme de l'image originale, e) Histogramme de l'image (b), f) Histogramme de l'image (c).*

pour l'IDC, figure 96.b est de 22690 octets. Cela correspond à 8.66% de la taille de l'image originale. La figure 96.c transporte un message de 3245 octets et a un taux de compression de 1.25.

5.3.4 Conclusion

Dans cette section, nous avons présenté une méthode réversible d'IDC combinée à du chiffrement et de la compression. Cette méthode est basée sur la décomposition de l'image, sur une compression basée sur RLE et sur la définition de l'OLLS. La méthode proposée est capable de cacher une information de taille égale à 8% de la taille de l'image originale. Dans ce cas, nous pouvons atteindre des taux de compression d'environ 1.2. Pour une compression optimale, nous pouvons insérer des données de l'ordre de 1.3% de la taille de l'image originale avec un taux de compression moyen de 1.35. Le taux de compression et la capacité d'insertion sont dynamiques. La méthode de chiffrement est basée sur des opérations de l'AES. Nous avons atteint une entropie très proche de 8 bits/pixel. La capacité d'IDC proposée est significativement plus importante qu'avec les méthodes classiques d'IDC dans le domaine spatial. Cette méthode propose un algorithme de chiffrement efficace et permet de reconstruire l'image originale sans aucune perte. Dans notre méthode, trois processus, l'IDC, la compression et le cryptage ont été groupés en une seule étape. Par conséquent cela diminue le temps de traitement et notre méthode est donc applicable sur des systèmes de faible puissance comme les téléphones mobiles par exemple.

5.4 Sécurisation de la HR d'une RI dans une image fortement comprimée

5.4.1 Introduction

Le transfert d'information sur les réseaux demande une forte compression des images. Il faut alors trouver un compromis entre rapidité de transfert et conservation d'une très bonne qualité de l'information. Généralement seules certaines régions de l'image présentent un intérêt majeur. De ce fait il n'est pas nécessaire de conserver une très haute qualité pour toute l'image [Wakatani 02, Wenjing 04, Gokturk 01, Strom 97]. Une fonctionnalité semblable est offerte par le futur standard de compression JPEG 2000 : il est possible de conserver des régions d'intérêt (RI) avec une haute résolution (HR) en compressant le reste de l'image [Christopoulos 00]. Dans cette section nous présentons une nouvelle méthode

qui inclue par insertion de données cachées (IDC) l'information perdue d'une RI lors de la compression JPEG d'une image. L'information perdue représente les pertes engendrées par l'algorithme de compression JPEG dans une RI de l'image. Cette information est incluse dans les coefficients DCT par un algorithme d'IDC [Chang 02] présenté section 3.3.2.3. Dans le cas du JPEG2000 la HR de la RI est visible dès l'ouverture de l'image comprimée. Dans le cas de notre méthode, la HR de la RI est protégée à l'ouverture de l'image comprimée [Amat 05]. En effet avec notre méthode, l'information complète de la RI n'est visible seulement qu'après l'extraction des données cachées et la reconstruction de la HR de la RI. La RI est une zone de l'image préalablement choisie. La RI peut avoir des formes et des tailles différentes selon les applications (plaque d'immatriculation, organe anatomique, visage de personne). Nous prenons une RI rectangulaire déterminée par les coordonnées du point supérieur gauche et sa taille afin de réduire son codage. La figure 97 illustre un exemple de RI. La taille de l'image originale est de 448×296 pixels et la taille de la RI est 112×48 pixels (5%). Cependant, nous verrons dans les résultats obtenus que nous pouvons augmenter la taille de la RI.

FIG. 97 – *Exemple d'une RI dans une image originale.*

La section 5.4.2 montre les différentes étapes de notre méthode de protection de données en présentant tout d'abord les modifications apportées à la méthode de [Chang 02], la récupération des pertes de HR dans la RI, la méthode d'IDC ainsi que la méthode d'extraction et la reconstruction de la HR dans la RI. Dans la Section 5.4.3, nous appliquons notre méthode à une image réelle.

5.4.2 Méthode de protection proposée

5.4.2.1 Adaptation de la méthode de [Chang 02]

Comme nous l'avons vu précédemment, la méthode de [Chang 02] a une capacité d'insertion qui est fixe. Pour notre problème, la quantité de données à insérer étant variable en fonction de la taille de la RI et du facteur de compression choisi, il a donc fallu adapter cette méthode. Dans un premier temps nous récupérons la totalité des pertes de la RI. En connaissant la quantité de bits du message à cacher, nous pouvons ensuite moduler le nombre de 1 dans la MQ (matrice de quantification). La figure 98 montre que ce processus a une phase d'optimisation, les modifications de la MQ affectant directement les pertes de données dans la RI. Cette méthode permet d'insérer une quantité de 1 plus optimale que dans la méthode initiale de [Chang 02]. De cette façon le taux de compression est meilleur et la qualité de l'image est plus proche de celle du JPEG original.

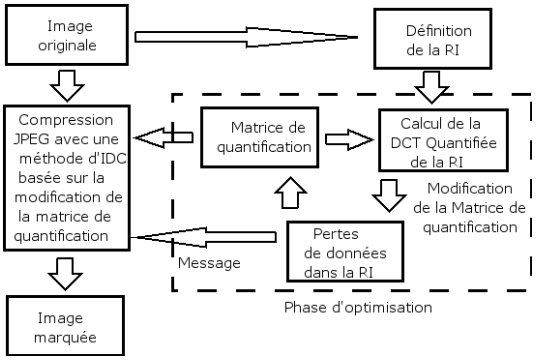


FIG. 98 – *La méthode proposée de protection de données.*

5.4.2.2 Récupération des pertes

Afin de proposer une méthode de reconstruction d'une RI sans perte, il a fallu s'intéresser aux parties de l'algorithme qui engendrent des pertes et à la façon de les récupérer. Les pertes provenant de la quantification ne pose pas de problème quant à leur récupération. Le problème principal vient du fait que certaines pertes proviennent de l'arrondi de l'IDCT lors de la phase de décompression. Il faut prendre en compte le fait que la méthode d'IDC engendre également des pertes. La figure 99 montre comment on arrive à calculer la totalité des pertes engendrées par la compression de l'IDC sur chaque bloc de la RI. Pour les pertes dûes a l'IDC on positionne à zéro tous les bits de poids faible dont on va se servir pour effectuer l'IDC (LSB0 et LSB1). De ce fait, lors de la phase de récupération des

données cachées on replacera les bits à zéro après lecture. Cela nous permet d'avoir une méthode de calcul des pertes qui ne dépend pas du message à marquer et aussi d'éviter que l'IDC engendre des erreurs supplémentaires. Avec cette méthode on s'assure que les pertes sont bien la différence entre la RI de l'image originale et la RI de l'image que l'on va reconstruire.

Les pertes sous cette forme ne nous permettent pas d'effectuer l'IDC. Etant donné qu'elles vont être insérées puis récupérées sous la forme d'un flux binaire, il est nécessaire de les mettre en forme pour faciliter leur extraction. La méthode choisie est la méthode de Huffman [Huffman 62] qui utilise des tables de préfixes pour encadrer les valeurs à coder. Cette méthode comporte un avantage dans notre cas car chaque section de coefficient sera codée avec le même nombre de bits. Ainsi il nous est d'autant plus facile, pendant la phase d'optimisation, d'évaluer la quantité de bits à tatouer sans avoir à effectuer le codage de Huffman. Les tables de préfixes utilisées sont celles inclues dans le JPEG.

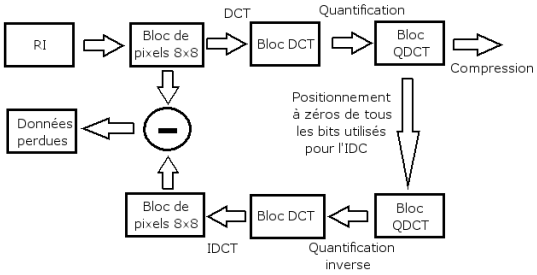


FIG. 99 – *Récupération des pertes.*

5.4.2.3 Modification de la matrice de quantification

La récupération des pertes, nous permet d'évaluer précisément la quantité de données perdues. Cette quantité va nous permettre de calculer exactement la quantité de 1 à insérer dans la matrice de quantification afin de pouvoir réaliser l'IDC. La figure 100 montre comment la MQ est modifiée afin d'avoir le nombre optimum de 1 nécessaire à l'IDC. Le fait d'insérer des 1 dans la MQ modifie la quantité de pertes de la RI. En effet, certains coefficients ne seront plus quantifiés. Cette opération est donc réalisée plusieurs fois afin que les pertes de données et le nombre de 1 dans la MQ soient optimums. Le résultat de cette optimisation nous concède ainsi la MQ qui sera utilisée pour exécuter la compression JPEG et l'IDC.

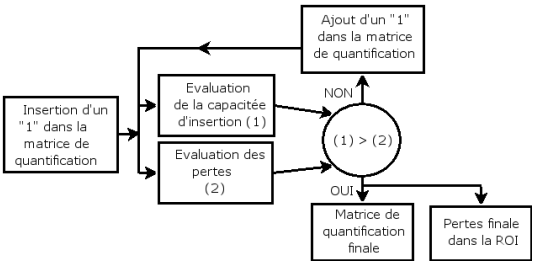


FIG. 100 – *Modification de la matrice de quantification.*

5.4.2.4 IDC

L'IDC va être effectuée sur les coefficients DCT qui seront quantifiés par 1 (Figure 39) [Chang 02]. L'IDC se fera par substitution des deux bits de poids faible LSB0 et LSB1. Il sera effectué dans tous les blocs de l'image même ceux contenus dans la RI, ceci est rendu possible grâce à la mise à zéro des coefficients non quantifiés.

5.4.2.5 Extraction et reconstruction

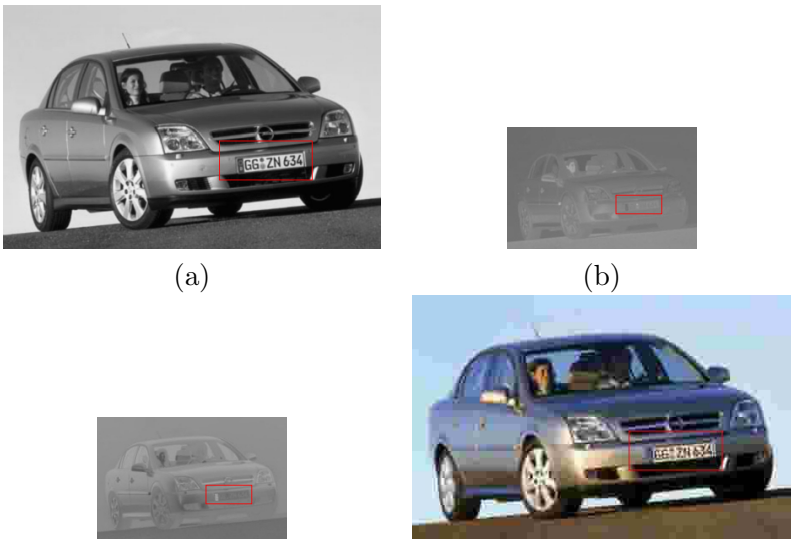
Cette section traite de l'extraction des données cachées et de la reconstruction de la RI. La reconstruction de la HR dans la RI se passe en deux étapes :

- La première étape consiste à récupérer les données cachées, pour cela il suffit de lire les bits de poids faible pour les coefficients dont la position dans le bloc correspond à un 1 dans la matrice de quantification (Celle-ci est contenue dans les entêtes du JPEG). Les premières données reçues sont la position et la taille de la RI. Tous les autres bits récupérés sont placés dans un vecteur binaire qui est ensuite décodé à l'aide de la table de Huffman afin de créer un nouveau vecteur de pertes. A la fin de cette opération nous obtenons une image décomprimée (mais ne contenant pas la HR dans la RI) et un vecteur de pertes.
- La deuxième étape est la reconstruction de la HR dans la RI. A l'aide du vecteur de pertes on reforme des matrices 8×8 représentant les pertes dans chaque bloc de la RI. Les pertes récupérées étant des pertes dans le domaine spatial, il suffit de les ajouter aux blocs décomprimés correspondant de la RI.

5.4.3 Résultats

L'image originale figure 97, a une taille de 448×296 pixels et la taille de la RI est 112×48 pixels (5%). Les premières étapes de la compression JPEG pour des images couleur sont

la décomposition en plan de luminance et de chrominance puis l'échantillonnage des plans de chrominance. Cet échantillonnage entraîne des pertes dans la RI. Les figures 101.a, b et c montrent respectivement le plan de luminance Y et les plans de chrominance Cr et Cb obtenus pour l'image originale. La figure 101.d illustre l'image comprimée avec JPEG pour un facteur de qualité de 10%, la colonne 1 du Tableau 22 montre le taux de compression et le PSNR obtenus pour cette image.



(a) (b) (c) (d)

FIG. 101 – *a) Composante de luminance de l'image (448×288), b) Composante de chrominance Cr de l'image (224×144), c) Composante de chrominance Cb de l'image (224×144), d) Image fortement comprimée avec FQ = 10%.*

Image	Comprimée	Comprimée et marquée	Reconstruite
Taux de compression	261	59	1
PSNR image (dB)	24.91	25.79	26.12
PSNR RI (dB)	25.13	28.3	43.5

TAB. 22 – *Taux de compression et PSNR.*

La figure 102 représente l'image comprimée et marquée avec notre méthode pour un facteur de qualité de 10%. Le Tableau 23 indique la quantité de bits mis à 1 dans les matrices de quantification (luminance et chrominance) ainsi que le nombre de bits que l'on a pu insérer dans chaque plan à l'aide de ces matrices. Les résultats obtenus pour la

Image	Y	Cr	Cb
Nbre de 1 dans MQ	10	9	9
Nbre bits insérés	35936	7340	7998

TAB. 23 – *Résultats obtenus par plan.*

FIG. 102 – *Image fortement comprimée et marquée.*

FIG. 103 – *Image décomprimée avec reconstruction de la HR de la RI.*

FIG. 104 – *Image de différence entre l'image originale et l'image reconstruite.*

compression avec notre méthode d'IDC sont visibles dans la colonne 2 du Tableau 22. Nous observons que le rapport de compression est moins significatif dans notre méthode, ceci est dû aux 1 insérés dans la matrice de quantification. Ces 1 changent également le facteur de qualité que nous avons choisi au début. La différence de qualité entre l'image comprimée avec l'algorithme du JPEG et celle comprimée avec notre méthode est inférieure à 1 dB. Ceci prouve que notre méthode d'insertion de données est imperceptible pour le système visuel humain. La figure 103 montre l'image décomprimée avec reconstruction de la RI. L'image reconstruite a un PSNR un peu supérieur à celui de l'image comprimée et marquée car la HR de la RI est reconstruite. Le PSNR de la RI après la reconstruction de l'image est de 43.5 dB, la RI est donc bien reconstruite en haute résolution. La figure 104 illustre la différence entre l'image originale et l'image décomprimée avec la RI reconstruite.

5.4.4 Conclusion et perspectives

Dans cette section, nous avons présenté une méthode permettant de conserver de manière secrète la HR d'une RI dans une image fortement comprimée avec JPEG. Elle améliore un service qui existe dans JPEG 2000 et qui est mis en application dans l'algorithme JPEG en rajoutant l'aspect protection. Elle est basée sur le travail de [Chang 02]. Nous avons amélioré leur méthode pour y inclure l'information d'une zone déterminée d'une image.

Nous travaillons actuellement sur des améliorations possibles de cette méthode. Actuellement il est possible avec cette méthode et pour le plan de luminance de récupérer la totalité des pertes et de les insérer afin d'avoir une RI infinie lors de la reconstruction. Nous étendons cette méthode afin de conserver la totalité des pertes sur l'ensemble des plans de luminance et de chrominance. Une autre extension possible est la répartition des pertes dans les plans, cette répartition permettrait de marquer plus fortement les plans de chrominance qui comportent moins d'informations. De cette façon on pourrait augmenter le taux de compression de notre méthode.

5.5 Crypto-compression par cryptage sélectif

5.5.1 Introduction

Classiquement il y a deux possibilités pour assurer la confidentialité et la protection des données transmises. La première possibilité s'appuie sur des structures hardware avec des

protocoles de communications, des pare-feux et des cryptages complets. Le principal avantage de cette approche est la complète transparence pour les utilisateurs. Les utilisateurs de ce type de protection ne font attention ni au cryptage ni à la sécurité. L'inconvénient de ce type de méthode est qu'elle est appliquée toujours de la même manière, quelque soit l'application et le niveau de sécurité souhaité. Du coup, en général, ce type de protection nécessite un serveur énorme et du matériel hardware spécifique. Ce type d'approche n'est pas possible avec un environnement faible puissance ou pour des machines portables dans des véhicules mobiles par exemple.

La seconde manière d'assurer la confidentialité est d'adapter le niveau de protection en fonction de l'application et du temps disponible. C'est dans cette seconde approche que nous trouvons le cryptage partiel ou sélectif où les utilisateurs peuvent appliquer une sécurité proportionnelle ou réglable en fonction du niveau de protection désiré [Norcen 03].

Le cryptage sélectif (CS) est une approche qui ne chiffre qu'une partie des données afin de diminuer le temps de calcul tout en assurant une certaine sécurité. Cette section présente une nouvelle méthode de cryptage sélectif pour des images médicales comprimées au format JPEG [Puech 05]. Cette méthode est basée sur le cryptage par AES de certains flux binaires issus du codage par Huffman. Les résultats de la méthode proposée présentent un gain de temps de calcul significatif tout en conservant le taux de compression et le flux binaire initial de JPEG.

Un nombre important d'applications peut se contenter d'un niveau inférieur à un cryptage complet en utilisant un cryptage partiel ou sélectif. En effet nous pouvons citer de nombreuses applications où par exemple des parties de l'image doivent être visibles pour autoriser une recherche et une classification de données. Des applications dans le domaine de la formation présentent des images qui doivent être partiellement visibles sans révéler complètement toute l'information. Les peintures numériques doivent être présentées sur Internet avec une qualité visible réglable. Le transfert de photos depuis des téléphones portables peut également se contenter d'un cryptage partiel pour assurer la confidentialité. C'est aussi le cas des images médicales prises depuis un appareil médical et devant être envoyées sur le réseau afin d'établir un diagnostic à distance. De plus, l'appareil d'acquisition d'images médicales peut se trouver dans une ambulance ou dans tout autre véhicule mobile, et dans ce cas la transmission est effectuée par l'intermédiaire de réseaux sans fil. Pour des raisons vitales, dans ce type d'applications, les images doivent être transmises rapidement et sûrement, et dans ce cas un cryptage partiel ou sélectif semblent être la meilleure solution (compromis temps/sécurité).

5.5.2 Travaux précédents

La confidentialité dans un environnement faible puissance est généralement assurée par des programmes de cryptage. Pour des applications de traitement d'images, il est toujours important d'essayer de minimiser le temps de calcul. Cependant, les implémentations logicielles des cryptages classiques sont souvent trop lentes pour traiter des images et des vidéos pour des systèmes commerciaux [Liu 03]. Le cryptage sélectif (CS) peut correspondre à des applications ne nécessitant pas un cryptage complet mais uniquement un cryptage des données essentielles et pertinentes. Cependant, la sécurité d'un CS est toujours comparée à celle d'un cryptage complet. La seule raison d'accepter ce schéma est la réduction importante du temps de calcul par rapport à un cryptage total. Un CS a pour but de protéger seulement les parties visuelles les plus importantes d'une image médicale. Donc, l'utilisation d'un CS nécessite une analyse de l'application médicale visée afin de pouvoir décider si c'est approprié et si l'on obtient bien la confidentialité souhaitée.

Malgré l'apparition du JPEG2000, le format JPEG est encore le format le plus utilisé pour la compression d'images. Il est également largement utilisé en traitement d'images, de l'industrie au médical [Pennebaker 93]. Actuellement, le format JPEG a été développé sur des quantités de cartes dédiées à la compression pour les caméras numériques, les téléphones portables, les scanners, les machines mobiles et les appareils médicaux d'acquisition d'images. Ces dispositifs existent déjà et sont opérationnels afin d'optimiser le format JPEG, mais l'aspect protection n'est pas souvent pris en compte. Beaucoup de méthodes de CS ont été créées avec une approche de cryptage pour des images codées par transformée en cosinus discrète (DCT).

- Tang [Tang 96] a proposé une technique appelée permutation zigzag applicable à des vidéos ou des images basées DCT. Bien que sa méthode offre plus de confidentialité, elle diminue le taux de compression.
- Droogenbroec et Benedett [Droogenbroeck 02] sont à l'origine d'une technique qui crypte un nombre sélectionné de coefficients AC. Dans leur méthode, les coefficients DC ne sont pas cryptés car ils portent une information visible importante mais sont hautement prédictibles. De plus, le taux de compression est constant (par rapport à la compression seule) et conserve le format du flux binaire. Par contre la compression et le cryptage sont fait séparément et par conséquent leur méthode prend plus de temps que la compression seule.
- Hebert *et al* [Fish 04] ont proposé une méthode telle que les données sont organisées

dans une forme de flux binaire réglable. Ces flux binaires sont construits avec les coefficients DC et quelques coefficients AC de chaque bloc et sont arrangés dans des couches en fonction de leur importance visuelle. Le cryptage partiel est alors effectué au niveau de ces couches.

– D'autres méthodes ont été développées spécifiquement pour les vidéos [Kunkelmann 98, Alattar 99, Qiao 98, Zeng 99, Cheng 00, Wen 02].

5.5.2.1 Les modes de JPEG

La première partie du format standard JPEG a été présentée section 3.3.1. Après quantification, les coefficients DCT sont lus dans un ordre prédéfini en zigzag en partant des basses fréquences et en terminant par les plus hautes fréquences, figure 105. Ensuite, cette séquence de coefficients quantifiés est utilisée par le codage entropique.

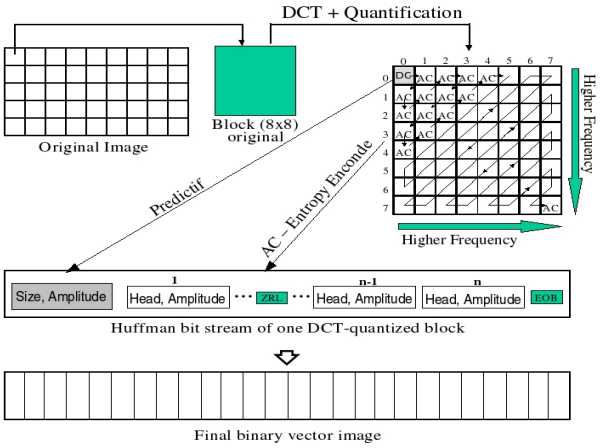


FIG. 105 – *L'algorithme JPEG.*

Le standard JPEG définit quatre systèmes de codage différents :

1. Système de codage en ligne
   - Processus basé DCT
   - Images uniquement sur 8 bits/pixel
   - Séquentiel
   - Codage de Huffman
2. Système basé DCT étendu
   - Processus basé DCT
   - Images uniquement sur 8 ou 12 bits/pixel

- Séquentiel ou progressif
- Codage de Huffman ou codage arithmétique

3. Système sans perte
   - Processus prédictif
   - Images de 2 bits à 16 bits
   - Séquentiel
   - Codage de Huffman ou codage arithmétique
4. Système hiérarchique
   - Processus basé DCT étendu ou sans perte
   - Multiples frames

Afin d'être compatible JPEG, l'algorithme de création doit inclure un support pour le système de codage en ligne [Gonzales 02]. Le système de codage en ligne est nécessaire pour tous les décodeurs DCT.

Dans le codage de Huffman les coefficients quantifiés sont codés par des couples {(HEAD),(AMPLITUDE)}. L'entête HEAD contient des contrôleurs obtenus par les tables de Huffman pour la compression et la décompression. Le paramètre AMPLITUDE est un entier signé correspondant à l'amplitude d'un coefficient AC non nul, ou dans le cas du coefficient DC de la différence entre deux coefficients voisins DC. La structure HEAD varie en fonction du type de coefficient. Pour les AC il est composé de (RUNLENGTH, SIZE), alors que pour les DC il est composé seulement de la taille SIZE, tableau 24.

DC	{ (SIZE)	AMPLITUDE }
AC	{ (RUNLENGTH, SIZE)	AMPLITUDE }

TAB. 24 – *Couple HEAD, AMPLITUDE pour les coefficients AC et DC.*

Les coefficients DC transportent une information visible importante et une corrélation locale significative. Ils sont hautement prédictibles, ainsi le codage de Huffman les traite séparément des 63 coefficients AC. La valeur des composants DC est importante et variée, mais est souvent très proche de celle de ses voisins. La seule valeur qui est donc encodée est la différence $DIFF$ entre le coefficient DC quantifié du bloc courant et le précédent DC_{i-1}. Les blocs sont lus de la gauche vers la droite, ligne par ligne :

$$DIFF = DC_i - DC_{i-1}.$$

La méthode présentée dans cet article est basée essentiellement sur le cryptage de certains coefficients AC. Pour cela, la description du codage de Huffman des coefficients AC est plus détaillée.

JPEG utilise une méthode alternative intelligente de codage des AC, basée sur la combinaison des informations longueur des séquences et amplitude, c'est-à-dire, qu'elle agrège les coefficients nuls quand il y a des plages de zéros. La valeur RUNLENGTH correspond au nombre de coefficients AC qui ont pour valeur zéro précédant une valeur non nulle dans la séquence en zigzag. La taille SIZE est la quantité de bits nécessaires pour représenter la valeur de l'amplitude. Afin de conserver une taille du tableau de codes inférieure à 256, la longueur de RUNLENGTH varie entre 0 et 15 et la taille SIZE entre 1 et 10 bits.

Deux codes particuliers correspondant à (RUNLENGTH, SIZE) = (0, 0) et (15, 0) sont utilisés pour symboliser la fin d'un bloc (EOB) et la longueur d'une plage de zéros. Le symbole EOB est transmis après le dernier coefficient non nul du bloc quantifié. C'est ainsi le chemin le plus efficace pour coder la fin d'une plage de zéros. Ceci peut être vu comme un symbole de sortie qui termine le bloc 8×8. Dans le processus de décodage, quand un symbole EOB est trouvé, tous les coefficients restants du bloc sont initialisés à zéro. Le symbole EOB est omis dans le cas où l'élément final du vecteur est non nul. Le symbole ZRL est transmis quand la valeur du RUNLENGTH est plus grande que 15 et il représente une longueur de plage de 16 zéros.

Nous donnons un exemple pratique, illustré tableau 25, qui montre un bloc original 8×8 de coefficients DCT quantifiés. Le tableau 26 présente le résultat du codage de Huffman. Si nous suivons le parcours en zigzag dans le bloc présenté tableau 25, le premier coefficient AC non nul est -5 sans valeur à zéro le précédant. Ceci produit une représentation intermédiaire de (0,3)(-5), où 3 est le nombre de bits nécessaire pour coder -5. Ensuite le coefficient AC suivant est 8, qui n'est pas non plus précédé de zéro. Par conséquent sa représentation intermédiaire est (0,4)(8). Les coefficients AC suivant sont deux zéros consécutifs suivis de la valeur 2, donc le (RUNLENGTH,SIZE)(AMPLITUDE) est (2,2)(2). La valeur du AC suivant à coder est -1 précédée de 3 zéros, donc sa représentation intermédiaire est (3,1)(-1). Après nous avons $RunLength > 15$, donc nous devons utiliser le symbole ZRL représentant 16 zéros successifs. Le dernier coefficient non nul est 1 précédé par un zéro, donc (1,1)(1). Comme c'est le dernier coefficient non nul, le symbole final de ce bloc 8×8 est la marque EOB.

Après avoir construit la représentation intermédiaire de Huffman, il est nécessaire de

97	-5	2	0	0	0	1	0
8	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

TAB. 25 – *Un bloc de coefficients DCT quantifiés.*

Représentation de Huffman intermédiaire		Flux binaire de Huffman	
HEAD	AMPLI.	HEAD	AMPLI.
(0,3)	-5	100	010
(0,4)	8	1011	1000
(2,2)	2	11111001	10
(3,1)	-1	111010	0
(ZRL)	—	11111111001	—
(1,1)	1	1100	1
(EOB)	—	1010	—

TAB. 26 – *Le flux binaire comprimé pour les coefficients AC du tableau 25.*

trouver le *Mot de code* pour la valeur de l'entête HEAD dans le tableau 27 et de calculer la représentation finale de l'amplitude AMPLITUDE en utilisant le tableau 28. Par conséquent, le mot de code pour (0,3) pris du tableau 27 est la séquence de bits 100. Pour (0,4) la séquence 1011 et ainsi de suite comme montré dans le tableau 26. La représentation finale de l'AMPLITUDE est calculée de la manière suivante. Le bit le plus à gauche est toujours utilisé pour le signe. Pour les valeurs négatives le bit de signe est égal à zéro et pour les valeurs positives à 1. Premièrement nous cherchons l'intervalle dans le tableau 28, pour (-5) par exemple, nous avons la taille en bits $k = 3$. Cela signifie que nous avons un bit pour le signe et deux pour représenter la valeur. Cette représentation est calculée par l'équation $V = |AMPLITUDE| - 2^{k-1}$. Si la valeur est négative le codage de Huffman utilise la notation avec le complément à 1 telle que tous les bits changent de valeurs. L'expression booléenne est $V = NOT(V)$. Pour -5 par exemple, nous avons $V = 5 - 2^2 = 1$, qui en binaire est 01. La valeur est négative donc le bit de signe est égal à 0, $V = NOT(01)$ et $V = 10$. La représentation finale en binaire de -5 est 010. Pour 8, nous avons $k = 4$ et par conséquence $V = 8 - 2^3 = 0$ et comme 8 est positif la représentation binaire finale est 1000 comme montré dans le tableau 26.

Run/Size	Longueur des codes	Mots de code
0/0 (EOB)	4	1010
0/1	2	00
0/2	2	01
0/3	3	100
0/4	4	1011
⋮	⋮	⋮
1/1	4	1100
1/2	5	11011
1/3	7	1111001
⋮	⋮	⋮
2/1	5	11100
2/2	8	11111001
⋮	⋮	⋮

TAB. 27 – *Mots de code pour les coefficients AC.*

Donc, pour chaque bloc 8×8, la séquence zigzag des 63 coefficients AC quantifiés est une séquence de bits qui peut être des paires de (HEAD, AMPLITUDE) ou des marques spéciales, EOB or ZRL. Avec l'exemple donné le résultat du flux binaire de Huffman est : 1000101011100011111001101110100...

Taille (k) en bits	Intervalles (-y..-x)	(x..y)
1	-1	1
2	-3,-2	2,3
3	-7...-4	4...7
4	-15...-8	8...15
5	-31...-16	16...31
6	-63...-32	32...63
7	-127..-64	64...127
8	-255...-128	128...255
9	-511...-256	256...511
10	-1023...-512	512...1023

TAB. 28 – *Taille du codage entropique pour les coefficients AC.*

Les tables de Huffman dans le format JPEG peuvent être adaptées (envoyées dans l'entête) ou être les tables par défaut. Le meilleur taux de compression est souvent obtenu avec la table par défaut du codage de Huffman car dans ce cas il n'est pas nécessaire d'insérer la table créée dans l'image comprimée. Par conséquent, pour obtenir des valeurs stables nous avons utilisé les tables par défaut du codage de Huffman.

5.5.2.2 L'algorithme de cryptage AES

En janvier 1997, l'institut NIST (U.S. National Institute of Standards and Technology) fait un appel à création d'une nouvelle méthode de chiffrement symétrique par bloc. L'AES (Advanced Encryption Standard) a pour objectif de remplacer le DES (Data Encryption Standard) qui devient vulnérable. Entre cinq candidats finalistes (MARS, RC6, Rijndael, Serpent, Twofish), le Rijndael soumis pas Joan Daemen et Vincent Rijmenwas a été déclaré vainqueur en octobre 2000. Le choix de Rijndael par rapport aux autres algorithmes finalistes est basé principalement sur son efficacité et son faible coût mémoire car il s'appuie sur l'utilisation de simples opérations binaires.

L'algorithme AES consiste en un ensemble d'étapes répétées un certain nombre de fois (rondes). Le nombre de rondes dépend de la taille de la clef et de la taille des blocs de données. Le nombre de rondes dans Rijndael est 9, si les blocs et la clef sont de longueur 128 bits. Il est de 11, si le bloc ou la clef est de longueur 192 bits, et que aucun d'eux n'est plus long que 192 bits. Le nombre de rondes est 13 si le bloc ou la clef est de longueur 256 bits. Pour crypter un bloc de données avec AES, figure 106, il faut d'abord effectuer l'étape nommée AddRoundKey qui consiste à appliquer un ou exclusif entre une sous clef et le bloc. Les données rentrantes et la clef sont donc additionnées ensemble dans la première étape AddRoundKey. Après, nous entrons dans l'opération d'une ronde. Chaque opération régulière de ronde implique quatre étapes.

La première est l'étape nommée "SubByte", où chaque octet du bloc est remplacé par une autre valeur issue d'une S-box. La seconde étape est l'étape nommée "ShiftRow" où les lignes sont décalées cycliquement avec différents offsets. Dans la troisième étape, nommée "MixColumn", chaque colonne est traitée comme un polynôme, multipliée sur GF(2^8) (Galois Field) par une matrice. La dernière étape d'une ronde est à nouveau l'étape nommée "AddRoundKey", qui est un simple ou exclusif entre la donnée actuelle et la sous clef de la ronde courante. L'algorithme AES effectue une routine supplémentaire finale qui est composée des étapes SubByte, ShiftRow et AddRoundKey avant de produire le chiffrement final.

Le processus sur les données en clair est indépendant de celui appliqué sur la clef secrète, et cette dernière est appelée Key Schedule. Celle-ci est formée de deux composantes : la Key Expansion et la Round Key Selection. La clef d'expansion est un tableau linéaire de mots de 4 octets et est notée $W[Nb * (Nr + 1)]$, où Nb est le nombre de colonnes du bloc de données et Nk est le nombre de colonnes de la clef de chiffrement. Les premiers mots

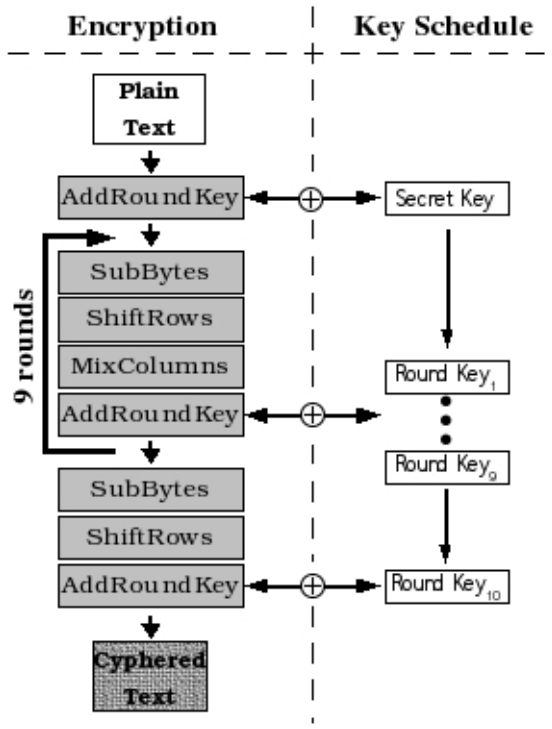


FIG. 106 – *Le schéma général d'AES.*

Nk contiennent la clef de chiffrement et tous les autres mots sont définis récursivement. La fonction d'expansion de la clef dépend de la valeur de Nk. La clef de chiffrement est étendue dans la Expanded Key. Les rondes de clefs sont prises pour la Expanded Key avec le chemin suivant : la première ronde de clef consiste en l'obtention des Nb premiers mots, la seconde ronde de clef consiste en l'obtention des Nb suivants et ainsi de suite [DR02, AES01].

L'algorithme AES peut supporter les modes de chiffrement suivants : ECB, CBC, OFB, CFB et CTR. Le mode ECB (Electronic CodeBook) est le mode de l'algorithme standard AES comme décrit dans la documentation 197 du standard FIPS (Federal Information Processing Standards).

A partir d'une séquence binaire X_1,X_2...X_n de blocs en clair, chaque X_i est chiffré avec la même clef secrète k afin de produire les blocs chiffrés Y_1,Y_2...,Y_n . Le mode CBC (Cipher Block Chaining) rajoute au chiffrement par bloc un mécanisme de retour. Chaque bloc chiffré Y_i est additionné par un ou exclusif avec le bloc clair rentrant X_{i+1} avant d'être crypté avec la clef k. Un vecteur d'initialisation (initialization vector IV) est utilisé pour le première itération. En fait tous les modes (sauf ECB) ont besoin d'un vecteur

d'initialisation IV. Dans le mode CFB (Cipher FeedBack) $IV = Y_0$. La clef dynamique Z_i est générée par $Z_i = E_k(Y_{i-1}), i \geqslant 1$ et le bloc chiffré est produit par $Y_i = X_i \oplus Z_i$. Dans le mode OFB (Output FeedBack), comme dans CFB, $Y_i = X_i \oplus Z_i$ mais $IV = Z_0$ et $Z_i = E_k(Z_{i-1}), i \geqslant 1$. Les données en entrée sont cryptées par un ou exclusif avec la sortie Z_i comme le montre la figure 107. Le mode CTR (Counter) a des caractéristiques très similaires à OFB, mais en plus il autorise une propriété d'accès aléatoire pour le décryptage. Il génère la clef dynamique suivante par cryptage de valeur successive d'un compteur. Ce compteur peut être une fonction simple qui produit une séquence pseudo-aléatoire. Dans ce mode, la sortie du compteur est l'entrée de AES.

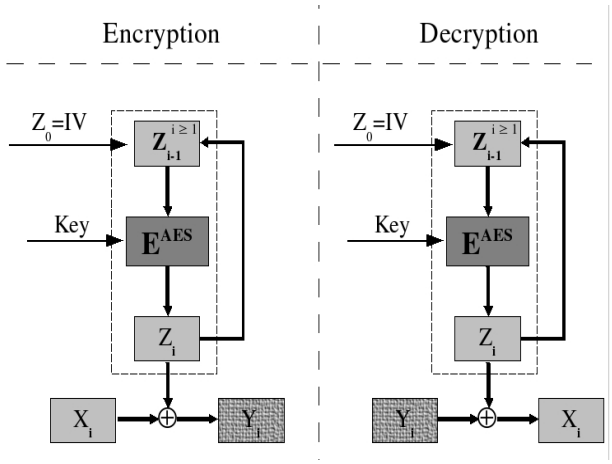


FIG. 107 – *Cryptage et décryptage pour le mode OFB.*

Même si AES est un algorithme de chiffrement par bloc, les modes OFB, CFB et CTR opèrent comme des chiffrements par flot. Ces modes ne nécessitent aucune mesure particulière concernant la longueur des messages qui ne correspond pas à une longueur multiple de la taille d'un bloc puisqu'ils travaillent tous en effectuant un ou exclusif entre le texte clair et la sortie du chiffrement par bloc. Chaque mode décrit a différents avantages et inconvénients. Dans les modes ECB et OFB par exemple tout changement dans le bloc du texte clair X_i provoque dans le bloc chiffré correspondant Y_i une modification, mais les autres blocs chiffrés ne sont pas affectés. D'un autre coté, si un texte clair du bloc X_i est changé dans les modes CBC et CFB, alors Y_i et tous les blocs chiffrés conséquent seront affectés. Ces propriétés signifient que les modes CBC et CFB sont utiles pour des problèmes d'authentification et les modes ECB et OFB traitent séparément chaque bloc. Par conséquent, nous pouvons noter que le mode OFB ne diffuse pas le bruit, alors que le mode CFB le diffuse.

A partir de la figure 107, il est important de noter que la fonction de cryptage $E_K^{AES}(X)$ est utilisé pour la phase de cryptage mais également pour la phase de décryptage dans le mode OFB.

5.5.3 La méthode proposée

Dans ce travail nous proposons une nouvelle méthode de cryptage sélectif pour des images médicales comprimées avec JPEG. Cette méthode est basée sur l'algorithme AES (Advanced Encryption Standard) en utilisant le mode de chiffrement par flot OFB (Output Feedback Block) dans l'étape du codage de Huffman de l'algorithme JPEG. La combinaison du cryptage sélectif et de la compression permet de gagner du temps de calcul et de conserver le format JPEG et le taux de compression initial. Pour le processus de décryptage le gain est amélioré de la même manière.

Soit $E_k(X)$ le cryptage d'un bloc X de n bits utilisant la clef secrète k avec l'algorithme AES en mode OFB. Dans la description de la méthode, nous supposerons $n = 128$ et X un texte clair non vide. Soit $D_k(Y)$ le décryptage d'un texte chiffré Y en utilisant la clef secrète k.

5.5.3.1 Procédure de cryptage

Le cryptage de la méthode proposée est appliqué en même temps que le processus de codage entropique durant la création du vecteur de Huffman. Cependant, notre méthode peut être appliquée sur tous les systèmes de codage JPEG utilisant la table de Huffman, décrite section 5.5.2.1.

L'idée principale de la méthode proposée est illustrée figure 108 et résumée ci-dessous :

1. Prendre les coefficients AC non nuls du flux binaire de Huffman, des plus hautes fréquences vers les basses fréquences afin de construire le vecteur du message en clair X.
2. Coder X avec l'algorithme AES en mode OFB.
3. Substituer le flux binaire de Huffman par l'information cryptée qui est de même taille.

Ces opérations sont appliquées séparément pour chaque bloc DCT quantifié.

Avant de présenter en détail la méthode, nous souhaitons prendre en compte quelques considérations.

– La raison de construire un chemin des hautes fréquences vers les basses fréquences

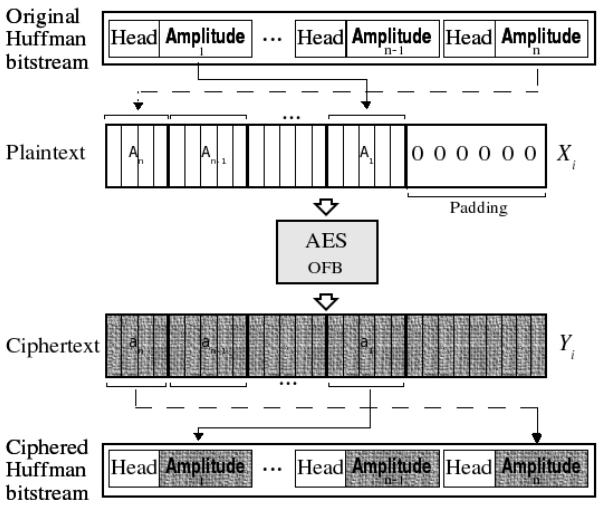


FIG. 108 – *Présentation générale de la méthode proposée.*

(ordre zigzag inverse) vient du fait que les caractéristiques visuelles les plus importantes de l'image se situent dans les basses fréquences, alors que les détails sont localisés dans les hautes fréquences. Le système visuel humain (SVH) est plus sensible aux basses fréquences qu'aux hautes fréquences. Cependant, nous pensons qu'il est intéressant de pouvoir calibrer l'apparence visuelle de l'image résultante. Cela signifie que nous nous orientons vers une méthode de cryptage réglable qui peut augmenter jusqu'à se rapprocher fortement de la composante DC de chaque bloc (basses fréquences).

– Le vecteur de Huffman est composé de couples {HEAD, AMPLITUDE} et de marques de contrôle ZRL et EOB. Ces marques de contrôle n'apparaissent pas obligatoirement, mais elles peuvent apparaître dans les cas suivants. Si les derniers coefficients AC dans le parcours en zigzag sont des zéros, le flux binaire de Huffman pour ce bloc doit contenir la marque EOB. La marque ZRL est trouvée chaque fois que seize zéros successifs sont rencontrés dans le parcours en zigzag et si il y a encore un coefficient AC non nul dans le bloc. Dans notre méthode de cryptage sélectif, nous ne modifions rien dans la partie HEAD ainsi qu'au niveau des marques de contrôles indiquées. Pour garantir une compatibilité totale avec tous les décodeurs, le flux binaire doit seulement être modifié dans les zones ou cela ne compromet pas les souhaits du format original JPEG.

– En codage, le bourrage (padding) est une méthode permettant d'additionner des

textes clairs de longueur variable. Ceci est nécessaire car le cryptage travaille sur une taille binaire fixée, mais la longueur du message en clair peut varier. Certains systèmes complexes de bourrage existent mais nous utiliserons le plus simple, en rajoutant des bits à zéros afin d'atteindre la longueur de bloc souhaitée. Historiquement, le bourrage est utilisé afin de rendre la cryptanalyse plus difficile, mais actuellement le bourrage est plus utilisé pour des raisons techniques avec les chiffrements par bloc, les fonctions de hachage et la cryptographie à clef publique.

– Une caractéristique concernant la quantité maximale de bits utilisés pour construire le texte clair X est à prendre en compte. Cette caractéristique règle le niveau de cryptage et la qualité visuelle de l'image résultat. Si rien n'est stipulée, la valeur du nombre de bits chiffrés est la taille du bloc chiffré $n = 128$. La taille du bloc est une contrainte dans le sens que nous ne pourrons pas chiffrer plus de $n = 128$ bits par bloc.
– Plus un bloc de l'image originale est homogène, plus il y a des zéros au niveau des coefficients AC quantifiés. En effet, la DCT (Discrete Cosine Transform) sépare l'image en sous-bandes spectrales. Donc les régions de l'image qui sont monotones fourniront des coefficients DCT proches de zéro qui après la quantification deviendront nuls [Yhang 00].

En détail, notre méthode travaille en trois étapes : la construction du texte clair X_i, le cryptage de X_i pour créer Y_i et la substitution du vecteur original de Huffman par l'information cryptée.

5.5.3.1.1 Construction du texte clair X Pour construire le texte clair X_i, nous prenons les coefficients AC non nuls du bloc courant i en accédant au vecteur de Huffman de la fin vers le début afin de créer des paires {HEAD, AMPLITUDE}. De chaque entête HEAD nous obtenons la longueur de l'AMPLITUDE en bit. Ces valeurs sont calculées à partir de l'équation 84. Comme montré dans la vue générale de la méthode proposée figure 108, seulement les AMPLITUDE (A_n,A_{n-1}...A_1) sont prises en compte pour construire le vecteur X_i. Le message en clair final L_{X_i} dépend à la fois de l'homogénéité ρ du bloc et de la contrainte donnée C. Cette contrainte C spécifie la quantité maximale de bits qui doit être prise en compte dans chaque bloc. D'un autre coté, l'homogénéité dépend du contenu de l'image et spécifie la quantité minimale de bits. Cela signifie qu'un bloc avec un grand ρ va produire un petit L_{X_i}. Le vecteur de Huffman est traité tant que $L_{X_i} < C$ et que

le coefficient DC n'est pas atteint. Ensuite, nous appliquons la fonction de remplissage (padding) $p(j) = 0$, où $n \geq j > L_{X_i}$, afin de remplir si nécessaire avec des zéros le vecteur X_i :

$$f(\rho) \quad \leq \quad L_{X_i} \quad \leq \quad C, \tag{84}$$

où :

$$\begin{cases} f(\rho) = 0, \quad \text{où} \quad \rho \to \infty \\ \\ \text{et} \qquad\quad C \in \{128, 64, 32, 16, 8, 4\} \text{ bits.} \end{cases}$$

5.5.3.1.2 Chiffrement de X avec AES en mode OFB Dans l'étape de chiffrement, le texte clair X_i est utilisé comme entrée pour le cryptage par AES afin d'obtenir Y_i. Le vecteur IV pour la première itération est créé à partir de la clef secrète k avec la stratégie suivante : La clef secrète k est utilisée comme une semence pour un générateur de nombres pseudo-aléatoire (GNPA). Ce k est divisé en 16 portions de 8 bits chacun. Le GNPA produit 16 nombres aléatoires qui définissent l'ordre de formation du vecteur IV. Par exemple si le premier nombre aléatoire généré est 7, le premier octet de la clef secrète sera copié dans le septième élément du vecteur IV. Si le second nombre aléatoire généré est 10, le second octet de la clef occupera le $10^{ème}$ octet dans le IV et ainsi de suite. Après avoir généré le vecteur $IV = Z_0$, il est premièrement chiffré afin de produire Z_1 puis chiffré pour produire Z_2, puis chiffré pour produire Z_3 et ainsi de suite, comme illustré figure 107. Ensuite chaque Z_i est additionné par un ou exclusif avec le texte en clair X_i pour générer Y_i.

5.5.3.1.3 Substitution du flux binaire de Huffman L'étape finale est la substitution de l'information initiale par l'information chiffrée dans le vecteur de Huffman. Comme dans la première étape (construction du texte clair X_i), le vecteur de Huffman est lu depuis la fin vers le début mais le vecteur chiffré Y_i est lu du début vers la fin. Connaissant la longueur en bits de chaque AMPLITUDE (A_n,A_{n-1}...A_1), nous commençons par couper ces portions dans Y_i pour remplacer l'AMPLITUDE dans le vecteur de Huffman. La quantité totale de bits doit être L_{X_i}.

Cette procédure est faite pour chaque bloc. Les blocs homogènes ne sont pas ou peu chiffrés. L'utilisation du mode OFB pour le chiffrement permet une génération de clef dynamique Z_i indépendante.

5.5.3.2 Haut niveau optimisé de chiffrement

Dans la section précédente nous avons présenté une méthode réglable pour chiffrer une image. Dans cette section nous présentons une approche pour optimiser un haut niveau de chiffrement (HNC). Celui-ci est basé sur les valeurs positives ou négatives données par le codage entropique de Huffman. Les valeurs positives ou négatives des AMPLITUDE subissent des traitements différents dans le codage de Huffman. Le fait de changer seulement le bit de signe de l'AMPLITUDE peut changer complètement la valeur dans le processus de décodage. Le codage de Huffman utilise la notation en complément à 1 pour les nombres négatifs et en exploitant cette caractéristique les valeurs sont complètement changées. Le nombre binaire '000', par exemple, est -7 pour l'AMPLITUDE dans la représentation de Huffman, alors qu'en changeant uniquement le bit de signe nous avons '100', la valeur -7 devient $+4$. Dans la première approche nous avons défini une contrainte C pour déterminer la quantité maximale de bits qui devait être prise en compte dans un bloc. Dans le chiffrement haut niveau optimisé tous les coefficients AC non nuls sont pris en compte. Les traitements sont illustrés figure 108. Le texte clair X est construit uniquement avec les signes de bits des coefficients AC non nuls, ce qui signifie que $L_{X_i} \leq 63$. Donc, le X est utilisé comme entrée pour le cryptage par AES pour générer Y. Après le cryptage de chaque bit de signe le vecteur de Huffman est remplacé par son chiffré correspondant comme décrit dans la section précédente.

5.5.3.3 Procédure de décryptage

La procédure de décryptage fonctionne de la manière suivante. Comme décrit précédemment, la clef secrète est utilisée pour construire le vecteur $IV = Z_0$. La valeur Z_1 est créée par cryptage de IV et la valeur Z_n est créée par cryptage de Z_{n-1}. De plus, les mêmes procédures que pour le cryptage, décrites dans la section précédente, sont utilisées. La différence est que l'entrée du processus de décryptage est le vecteur de Huffman chiffré. Le vecteur chiffré est aussi traité de la fin vers le début pour construire le texte en clair Y_i. La valeur Y_i est utilisée avec Z_i dans la procédure de décryptage par AES, illustré figure 107. Le vecteur résultat du texte en clair est coupé en parties afin de remplacer les AMPLITUDE dans le chiffré de Huffman pour de générer le vecteur de Huffman.

5.5.3.4 Exemple pratique

5.5.3.4.1 Chiffrement AC réglable Dans cette section un exemple pratique est présenté sur un bloc DCT quantifié. Comme le cryptage sélectif réglable travaille par

bloc, celui-ci peut être étendu sur toute l'image. Soit le bloc DCT quantifié du tableau 29.

243	-28	7	3	1	-6	3	-9
5	-2	-7	-2	0	-4	4	-9
-2	1	2	1	-1	2	-3	3
-1	0	2	1	1	0	2	2
1	0	-3	-1	-1	-1	-1	-3
1	0	1	0	1	1	0	1
-3	0	1	1	0	0	0	1
3	0	-2	-1	0	0	0	-2

TAB. 29 – *Bloc original de coefficients DCT quantifiés.*

La représentation intermédiaire de Huffman est une séquence de paires de symboles {(Runlength, Size), AMPLITUDE} suivant le parcours en zigzag. Pour ce bloc, tableau 29, nous avons :

{(0,5),-28} {(0,3),5} {(0,2),-2} {(0,2),-2} {(0,3),7} {(0,2),3} {(0,3),-7} {(0,1),1} {(0,1),-1} {(0,1),1} {(1,2),2} {(0,2),-2} {(0,1),1} ⋯ {(1,2),3} {(1,1),1} {(1,1),-1} {(1,2),-3} {(0,4),-9} {(0,2),3} {(0,2),2} {(0,1),-1} {(0,1),1} {(0,1),1} {(0,2),-2} {(0,1),-1} {(1,1),1} {(0,1),-1} {(0,2),2} {(0,2),-3} {(5,1),1} {(0,1),1} {(1,2),-2}.

Pour construire le texte en clair X nous accédons aux paires (RunLength, Size), AMPLITUDE dans l'ordre inverse et appliquons les tableaux 27 et 28. Notre méthode ne change pas l'information de l'entête HEAD, donc nous récupérons que l'information à changer. Le vecteur X suivra la transformation détaillée tableau 30.

Après conversion par le tableau 30, le texte clair complet X est : 0 1 1 1 0 0 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0 0 1 1 0 1 0 0 1 1 0 1 1 0 1 0 1 0 0 0 1 1 1 1 1 0 1 0 1 1 0 1 0 0 0 1 1 + PADDING(0). Dans ce cas la fonction de bourrage (padding) remplit le vecteur du texte en clair avec 40 zéros.

Original	Décimal	-2	1	...	5	-28
	Binaire	01	1	...	101	00011
Chiffré	Binaire	00	0	...	001	01101
	Décimal	-3	-1	...	-5	-18

TAB. 30 – *Conversion des AMPLITUDE décimal en binaire, cryptage, puis conversion en décimal dans la représentation de Huffman.*

Après cryptage avec AES en mode OFB, nous obtenons le vecteur crypté Y égal à **0 0**

0 0 0 1 0 1 0 1 1 0 1 0 0 1 1 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 1 1 1 1 0 1 0 1 1 0 1 0 1 1 1 0 1 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0 1 0 1 1 0 1 0 0 1 1...

243	-18	4	-2	1	4	3	10
-5	-3	5	2	0	-5	7	10
2	-1	-3	1	1	-3	2	-2
-1	0	-2	1	-1	0	2	-2
-1	0	2	-1	1	1	-1	-2
-1	0	-1	0	-1	1	0	-1
3	0	-1	-1	0	0	0	-1
2	0	-2	1	0	0	0	-3

TAB. 31 – *Bloc chiffré.*

5.5.3.4.2 Haut niveau de chiffrement Le haut niveau de chiffrement optimise le cryptage des coefficients AC non nuls dans le bloc en utilisant les bits de signe. Le processus suit les trois étapes (construction du texte en clair X_i, cryptage de X_i et substitution du vecteur original de Huffman par l'information chiffrée) comme décrit section 5.5.3.1.

Le tableau 32 montre dans la première ligne l'information originale en représentation décimale. Dans la seconde ligne l'information est en binaire suivant la représentation binaire du codage entropique par Huffman. La troisième ligne montre le changement binaire du signe et la dernière ligne montre les résultats obtenus en décimal.

Original	Décimal	-2	1	...	5	-28
	Binaire	01	1	...	101	00011
Signe	Binaire	11	0	...	010	10011
crypté	Décimal	3	-1	...	-6	19

TAB. 32 – *Modification des bits de signe des AMPLITUDE.*

En prenant le bloc original, représenté tableau 29, et en suivant le processus de traitement du bit de signe décrit section 5.5.3.2 et présenté tableau 32, nous obtenons le bloc résultat montré tableau 33.

5.5.4 Résultats expérimentaux

Pour toutes nos expériences, nous avons utilisé l'algorithme JPEG avec le système de codage en ligne séquentiel avec un facteur de qualité (FQ) de 100%. Nous avons appliqué sur les images cinq valeurs pour la contrainte C (128,64,32,16 et 8) et l'algorithme de haut

243	19	7	3	1	-6	3	14
-6	-2	-7	3	0	7	4	14
-2	1	-3	-1	1	2	2	-2
1	0	-3	1	-1	0	2	2
-1	0	2	1	-1	-1	1	2
1	0	1	0	-1	-1	0	-1
2	0	1	1	0	0	0	-1
-2	0	-2	1	0	0	0	3

TAB. 33 – *Bloc chiffré par haut niveau de chiffrement.*

niveau de chiffrement (HNC). Pour le chiffrement, nous avons employé l'algorithme AES avec le mode de chiffrement par flot OFB et avec une clef de longueur 128 bits. Cependant, notre méthode peut être employée avec d'autres valeurs de longueur pour la clef et pour les blocs.

	Information cryptée				
C	Coeffi cients	Bits	% Bits	% pixels changés	PSNR (dB)
128	26289	81740	23.0	85.7	23.39
64	23987	71900	20.2	85.7	24.42
32	18035	52101	14.6	85.3	25.02
16	10966	31106	8.8	83.5	27.66
8	6111	16765	4.7	76.1	30.90
HNC	29300	29300	8.2	84.0	21.21

TAB. 34 – *Résultats pour l'image rayons X d'un cancer du colon, figure 109.a,* 320×496 *pixels.*

Les méthodes ont été appliquées sur plusieurs dizaines d'images médicales en niveau de gris. Cependant, nous présentons les résultats tableaux 34 et 35 pour deux images médicales différentes, illustrées figures 109 et 110.

L'image médicale originale, de taille 320×496 pixels, comprimée ainsi que toutes les images cryptées ont la même taille, soit 43.4 Ko. Dans le tableau 34, on peut noter que les 2480 blocs 8×8 de l'image ont été changés. Cela signifie qu'il n'y a aucun bloc totalement homogène. Pour $C = 128$, maximum de 128 bits chiffrés par bloc, nous avons eu 26289 coefficients AC chiffrés et 81740 bits chiffrés, ce qui fait une moyenne de 33 bits chiffrés par bloc. Le pourcentage de bits chiffrés dans l'image entière est de 22.99% et ceci nous donne dans le domaine spatial 136038 pixels changés, ce qui correspond à 85.71% des pixels chiffrés. Le pic du rapport signal à bruit (PSNR) est de 23.39 dB pour $C = 128$.

Pour $C = 64$ la quantité de coefficients AC et de bits codés est respectivement de 23987 et de 71900. Le pourcentage de bits chiffrés par rapport à l'image entière est de 20.22%. Cette contrainte nous donne un nombre de pixels modifiés de 135959 qui correspond à 85.66% de tous les pixels de l'image. Le PSNR est alors de 24.42 dB. Pour le HNC nous atteignons 29300 coefficients cryptés. Comme tous les signes des coefficients AC ont été chiffrés, nous avons la même quantité de bits cryptés 29300. Alors que seulement 8.24% de bits de l'image ont été cryptés nous avons 133277 pixels changés dans l'image, soit 83.97% de toute l'image. Le PSNR est de 21.21 dB.

Dans le tableau 35 nous montrons le résultat de notre méthode appliquée sur une image médicale d'un scanner CT de taille 512×512 pixels. L'image originale après compression classique par JPEG et les images cryptées ont toutes la même taille, soit 59.9 Ko. Pour la contrainte $C = 128$, nous avons chiffré 51147 coefficients AC et 131127 bits, ce qui correspond à 32 bits par bloc en moyenne. Le pourcentage de bits chiffrés dans l'image entière est de 26.72%, ce qui nous donne 230424 pixels changés, soit 87.90% des pixels. Le PSNR est de 28.18 dB. Pour $C = 8$ la quantité de coefficients et de bits chiffrés est respectivement de 9633 et de 26606. Seulement 5.42% des bits de l'image sont chiffrés. Pourtant, avec $C = 8$, nous avons quand même 195821 des pixels de l'image qui sont modifiés, ce qui correspond à 74.68% de tous les pixels. Le PSNR est alors de 33.06 dB. Pour la méthode par HSC, nous atteignons 51576 coefficients changés et également la même quantité de bits changés, soit 10.51% des bits de l'image qui ont subis un cryptage. Nous avons avec cette méthode,230922 pixels changés, soit 88.09% des pixels de l'image entière. Le PSNR est 27.95 dB. Il est à noter que seul 10.51% des bits ont été cryptés alors que 88.09% des pixels de l'image ont perdu leur valeur initiale.

	Information cryptée				
C	Coefficients	Bits	% Bits	% pixels changés	PSNR (dB)
128	51147	131127	26.7	87.9	28.18
64	47656	119423	24.3	87.9	28.31
32	37995	95850	19.5	87.5	29.15
16	18957	53083	10.8	85.0	30.45
8	9633	26606	5.4	74.7	33.06
HNC	51576	51576	10.5	88.1	27.95

TAB. 35 – *Résultats pour l'image médicale scanner CT,* 512×512 *pixels.*

Comme nous pouvons voir sur les images résultats, le cryptage sélectif sur toute l'image

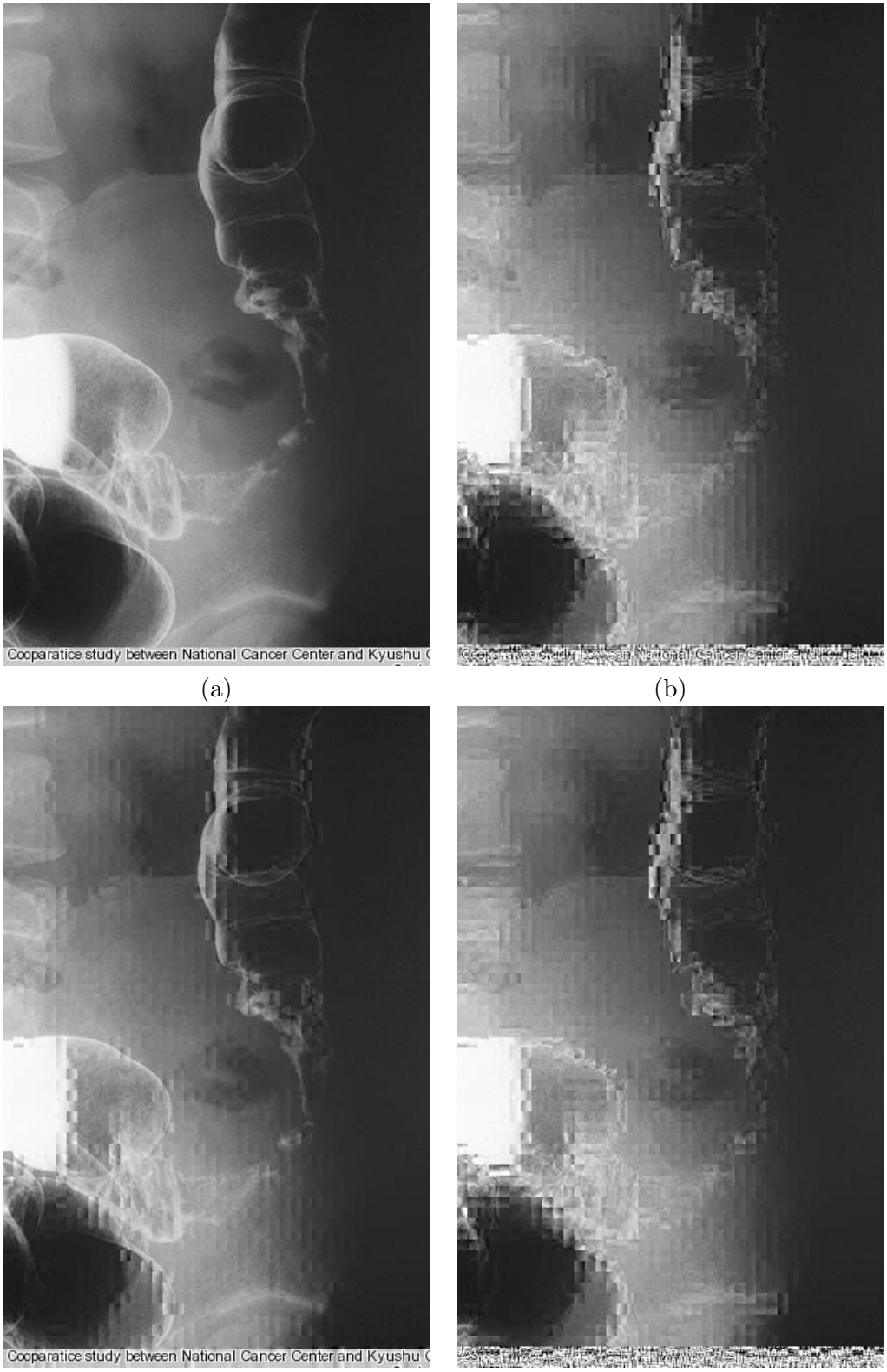
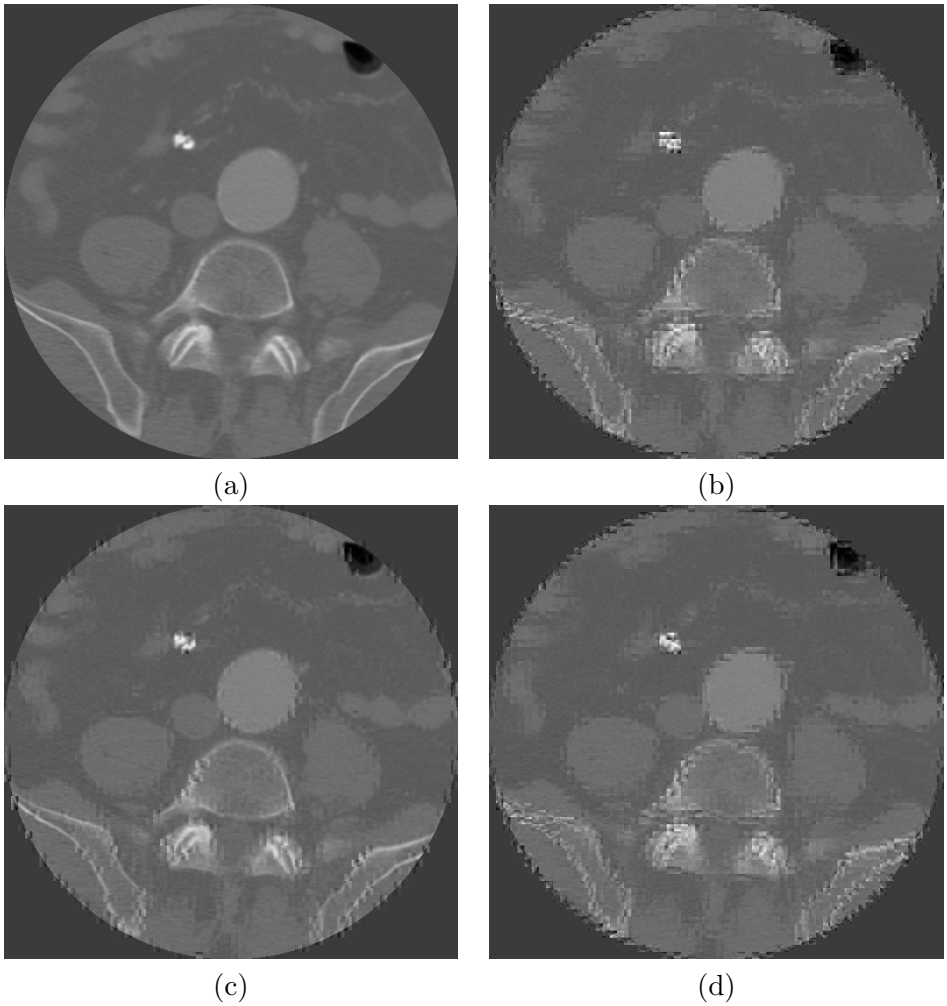


FIG. 109 – *a) Image médicale originale d'un cancer du colon,* 320×496 *pixels, b) Image cryptée pour* $C = 128$*, c) Image cryptée pour* $C = 8$*, d) Image cryptée HNC.*



(a) (b)

(c) (d)

FIG. 110 – *a) Image médicale d'un scanner* 512×512 *pixels, b) Image cryptée pour* $C = 128$*, c) Image cryptée pour* $C = 8$*, d) Image cryptée HNC.*

JPEG produit des artefacts par bloc. Ces artefacts sont au niveau des frontières des blocs, qui importunent souvent le SVH. Puisque la transformation fréquentielle et la quantification des blocs de pixels sont traitées séparément, la continuité des valeurs des pixels de blocs voisins est cassée durant le codage.

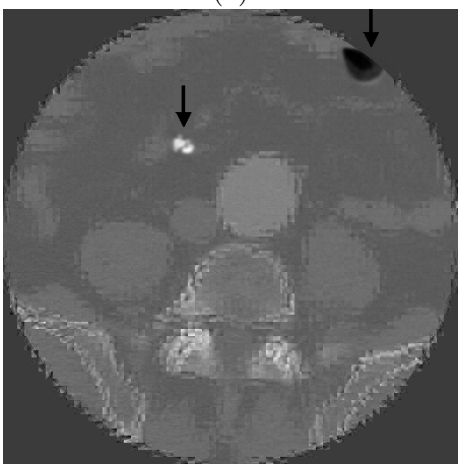
Un des avantages de notre méthode est la possibilité de décrypter de manière individuelle les blocs 8×8 pixels de l'image. Ceci est dû au fait que nous avons utilisé le mode par flot OFB pour le cryptage par AES. Les figures 111 montrent le décryptage partiel des images sur des régions d'intérêt. Dans ces exemples, les images peuvent être à 100% déchiffrées, mais chaque région peut être déchiffrée de manière réglable avec $C = 16$ ou $C = 32$ par exemple. Il est important de noter que la région de l'image qui est à décrypter doit être définie dans des tailles de blocs unitaires de 8×8 pixels, qui est la taille par défaut des blocs du JPEG. Dans la figure 111.a, une région de 13×9 blocs (soit 104×72 pixels) a été décryptée dans une région particulière. Au niveau de la figure 111.b nous avons décrypté deux régions particulières dans l'image de taille 40×48 pixels pour celle située à gauche et 64×64 pixels pour celle située à droite.

Il convient de noter que la sécurité est liée à la capacité de deviner les valeurs des données chiffrées (cryptanalyse). Par exemple, d'un point de vue de la sécurité, il est préférable de chiffrer les bits qui semblent les plus aléatoires. Cependant, en pratique, le remplacement des valeurs des coefficients AC non nuls est plus difficile que les valeurs des coefficients de DC d'une image JPEG qui sont fortement prévisibles [Droogenbroeck 02].

5.5.5 Conclusion

Dans cette section, nous avons proposé un nouveau schéma de cryptage sélectif pour des images médicales comprimées par JPEG en utilisant le cryptage AES en mode par flot OFB. Nous pouvons lister les avantages de notre méthode tels que la portabilité, un taux de compression constant, une compatibilité avec le format JPEG, un cryptage sélectif réglable en quantité et un décryptage partiel par région d'intérêt. Les résultats appliqués sur des images médicales ont montré que de notre méthode il résulte des PSNR masquant bien l'information (PNSR $< 30\ dB$) . Par rapport au pourcentage de bits et de pixels cryptés notre méthode fournit un niveau de confidentialité acceptable pour le transfert d'images médicales avec visualisation rapide à distance en temps réel.

(a)



(b)

FIG. 111 – *Décryptage partiel des images a) Décryptage d'une région, b) Décryptage de deux régions.*

5.6 Conclusion et perspectives

Dans ce chapitre nous avons présenté quatre nouvelles méthodes de protection de données combinant IDC, cryptage et compression. Les perspectives de mon travail s'orientent dans l'évolution de ces nouvelles méthodes hybrides. En effet, afin d'être performant au niveau du temps de calcul pour des transferts rapides il est nécessaire de coupler au moins deux de ces trois traitements.

Conclusion et perspectives

Chapitre 1

Conclusion générale

Ce document, composé de 2 parties, résume mon activité professionnelle depuis la soutenance de ma thèse. Ce manuscrit m'a permis de faire un bilan complet sur mes recherches. La protection du transfert des images sur Internet n'est actuellement pas encore résolu et dépend fortement du contexte de l'application.

Dans la première partie de ce document j'ai présenté en détails mon parcours en détaillant mes responsabilités pédagogiques et administratives ainsi que mes projets en cours. J'ai également présenté dans cette partie les étudiants que je co-encadré.

Dans la seconde partie de mon document j'ai présenté un résumé de 8 ans de travail que j'ai effectué en collaboration avec d'autres chercheurs et des étudiants. Ma thématique scientifique a évolué avec ma mobilité géographique. Cependant, j'ai essayé au maximum de garder une continuité dans mes travaux de recherche. Cette cohérence continue actuellement avec de nouveaux doctorants qui travaillent dans le domaine de la sécurité. En effet, l'aspect sécurité est actuellement la préoccupation principale de mes activités de recherche. Alors que dans le domaine de la compression des images et des vidéos les standards se sont imposés depuis de nombreuses années, pour la partie chiffrement et marquage il n'existe pas encore actuellement de normes. Je compte passer encore plusieurs années sur cette thématique en espérant pouvoir diriger des doctorants dans ce domaine.

Du fait de ma mobilité géographique, j'ai multiplié mes contacts et enrichi mes connaissances puisque depuis ma thèse j'ai travaillé dans 5 laboratoires de recherche. Cette mobilité m'a permis également de prendre du recul vis à vis de mes travaux et d'observer différentes approches de la recherche.

Enfin, j'ai eu à encadrer de nombreux étudiants dans le cadre de leur thèse ou de leur stage de DEA. Dans ce document, j'ai montré que la recherche est un métier d'équipe.

Dans la seconde partie de ce document, j'ai en particulier :

- fait un point sur les utilisations des images dans le domaine médical et montré que l'image pour le télédiagnostic était importante,
- présenté une nouvelle méthode de détection de contours par contours actifs dans une séquence d'images médicales,
- proposé une nouvelle méthode d'insertion de données cachées robuste à la compression tout en assurant une qualité maximale de l'image,
- fait un bilan complet entre le cryptage et l'image,
- proposé des méthodes hybrides associant compression, insertion de données cachées et cryptage d'image.

Chapitre 2

Perspectives

Dans ce chapitre, je présente la continuité de mes recherches présentée partie II en décrivant des axes de recherche que j'envisage de poursuivre. Actuellement je co-encadre trois doctorants au LIRMM dans les domaines de la sécurisation de transfert d'images et d'objets 3D. Ma thématique s'oriente vers la protection d'objets multimédias tels que les images, les objets 3D et les vidéos.

2.1 Numérisation sécurisée d'objets 3D

Après la phase de numérisation, un modèle numérique 3D doit être souvent transféré via Internet entre plusieurs intervenants afin d'être industrialisé. Les droits d'auteurs ne sont alors plus assurés. En plus des droits d'auteurs, rentrent en compte les problèmes d'authenticité ainsi que les problèmes d'intégrité. Actuellement, toutes les personnes qui ont accès au modèle numérique 3D peuvent utiliser celui-ci pour le revendre par exemple. Aucune information concernant les auteurs, la date et le lieu de création, ainsi que les techniques employées, n'est associée au modèle 3D. Les points durs de ces travaux sont, après l'obtention d'un système de mise en correspondance de texture avec un modèle 3D numérique, de trouver un système de sécurisation de formes 3D par insertion de données cachées et par cryptage multi résolution.

Pour le développement de modules de sécurisation, ce travail peut se décomposer en plusieurs parties qui sont :

- Analyse et extraction des données pertinentes caractéristiques de l'objet 3D à partir de modèles 3D déjà existants.
- Evaluation de la quantité de données à insérer de manière invisible et indélébile.
- Approche spatiale : topologie et forme.

- Approche fréquentielle.
- Normalisation avec les formats standards.
- Conversion multi-niveaux.
- Cryptage symétrique.
- Analyse de l'application des algorithmes classiques de cryptage.
- Cryptage multi-niveaux afin d'assurer une cohérence au niveau des accès.

2.2 Cryptage multirésolution d'images haute résolution

Dans le cadre d'une collaboration future avec les musées de France, nous souhaitons développer un système de sécurisation des bases de données de peintures numériques. L'objectif de ces travaux est de mettre en place une méthode laissant visible sur Internet la basse résolution d'une œuvre numérisée portant elle-même la haute résolution mais chiffrée. Nous pensons étendre cette approche à plusieurs niveaux de résolution.

La problématique de ces travaux de recherche consiste à obtenir :

- d'une part plusieurs niveaux de représentation de la même image,
- et d'autre part un certain nombre de clefs de cryptage hiérarchisées.

En effet, chaque clef dans la hiérarchie, autorisera l'accès à un niveau particulier de résolution de la peinture numérique.

2.3 Cryptage partiel et sélectif de régions d'intérêt dans des images médicales

Tout au long de ce document nous avons vu que l'usage des nouvelles technologies de l'information modifie l'environnement de travail des professionnels de santé sous la pression de plusieurs facteurs. Ceci est une exigence croissante des patients qui souhaitent une amélioration de leur prise en charge globale. Les professionnels souhaitent également assurer un meilleur suivi de leur patients et bénéficier des aides que peuvent apporter ces nouveaux outils des impératifs d'amélioration de la qualité et de l'efficacité économique du système de soins.

Différentes technologies dont l'insertion de données cachées et le cryptage sont utilisées à ce jour pour sécuriser le transfert des images médicales. Les techniques à développer doivent être robustes à toutes les attaques sans dégrader la qualité de l'image à la réception. L'image médicale représente une quantité volumineuse de données. L'utilisation des techniques dites asymétriques est fastidieuse en terme de temps de calcul. Les travaux réalisés

dans le domaine se basent sur les techniques symétriques qui s'adaptent mieux à ce type de données mais qui posent un problème d'intégrité de l'image mais également un problème de confidentialité dans le cas des images contenant des zones homogènes. Aussi, il est indispensable de définir un protocole d'échange de clés. Une autre alternative est le chiffrement par flot déjà exploité dans mes travaux.

L'objectif de ces travaux est de contribuer au domaine de la sécurisation du transfert des images médicales. Nous nous intéresserons à un cryptage sélectif (partiel) des régions d'intérêt (RI) de l'image médicale pouvant s'intégrer dans un schéma de compression JPEG 2000. Un cryptage hybride peut être alors envisagé selon l'importance des régions d'intérêt définies sur l'image. Afin de renforcer la sécurité des transmissions il nous semble important d'associer l'insertion de données cachées aux algorithmes de cryptage. L'insertion de données cachées dans une image pourrait nous aider à assurer la confidentialité et l'intégrité des informations textuelles associées à l'image relatives au patient.

Les étapes de réalisation de ces travaux de recherche sont décomposées en trois parties :

- Proposition d'un algorithme de cryptage sélectif basé sur les ondelettes dans un schéma de compression JPEG2000.
- Amélioration de l'algorithme de cryptage sélectif dans le domaine des ondelettes par utilisation des RI dans un schéma de compression JPEG2000.
- Développement de méthodes d'insertion de données cachées dans une image concernant les informations textuelles associées à l'image relatives au patient.

Bibliographie

[Abrantes 93] A.J. Abrantes and J.S. Marques. A New Algorithm for Active Contours. *Image Processing Theorie and Application*, 1993.

[AES01] AES. Announcing the Advanced Encryption Standard. *Federal Information Processing Standards Publication*, 2001.

[Alattar 99] A.M. Alattar, G.I. Al-Regib, and S.A. Al-Semari. Improved Selective Encryption Techniques for Secure Transmission of MPEG Video Bit-Streams. In *ICIP 99, International Conference in Image Processing, IEEE*, vol. 4, 256–260, 1999.

[Amadieu 99] O. Amadieu, E. Debreuve, M. Barlaud, and G. Aubert. Inward and Outward Curve Evolution Using Level Set Methods. In *Proc. International Conference on Image Processing, (ICIP-1999), Kobe, Japan*, 1999.

[Amat 05] P. Amat and W. Puech. Transfert sécurisé d'une RI sans perte par une méthode d'insertion de données cachées robuste à la compression JPEG. In *20th. Colloque Traitement du Signal et des Images (GRETSI'05), Louvain-la-Neuve, Belgique*, september 2005.

[Barbaresco 97] F. Barbaresco, S. Bonney, J. Lambert, and B. Monnier. Contours actifs géodésiques et à modèles contraints pour le suivi des orages dans un contexte multisenseur : radar, interférométre VHF, satellite IR. In *Proc. 16th. Colloque Traitement du Signal et des Images GRETSI'97, Grenoble, France*, pp. 717–720, 1997.

[Bas 98] P. Bas, J.M. Chassery, and F. Davoine. Tatouage d'images par modification du code fractal. In *Proc. 4th Colloque Compression et Représentation des Signaux Audiovisuels, (CORESA'98), Lannion, France*, 1998.

[Bas 01] P. Bas and B. Macq. A New Video-Object Watermarking Scheme Robust to Object Manipulation. In *Proc. of ICIP'01, Tessaloniki, Greece*, 526–529, 2001.

[Bas 02] P. Bas, J.M Chassery, and B.Macq. Image watermarking: an evolution to content based approaches. *Pattern Recognition*, 35, pp. 545–561, 2002.

[Bender 96] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for Data Hiding. *I.B.M. Systems Journal*, 35(3-4), pp. 313–336, 1996.

[Bernarding 01] J. Bernarding, A. Thiel, and A. Grzesik. A JAVA-based DICOM server with integration of clinical findings and DICOM-conform data encrytion. *International Journal of Medical Informatics*, 64, pp. 429–438, 2001.

[Berthilsson 97] R. Berthilsson, K. Astrom, and A. Heyden. Reconstruction of 3D-Curves from its 2D-Images Using Affine Shape Methods for Curves. *J. of Mathematical Imaging and Vision*, 1997.

[Borie 02a] J.C. Borie, W. Puech, and M. Dumas. Encrypted Medical Images for Secure Transfer. In *ICDIA 2002, Diagnostic Imaging and Analysis, Shanghai, R.P. China*, 250–255, Aug. 2002.

[Borie 02b] J.C. Borie, W. Puech, and M. Dumas. Encrypted Images for Secure Transfer with RSA Algorithm. In *IEEE Communication 2002, Bucharest, Romania*, Dec. 2002.

[Borie 04a] J.C. Borie, W. Puech, and M. Dumas. Crypto-Compression Using TEA's Algorithm and a RLC Compression. In *Proc. 2nd Intelligent Access to the Multimedia Documents on the Internet, MediaNet'04, Tozeur, Tunisia*, 5–16, Nov. 2004.

[Borie 04b] J.C. Borie, W. Puech, and M. Dumas. Crypto-Compression System for Secure Transfer of Medical Images. In *Proc. 2nd International Conference on Advances in Medical Signal and Information Processing, MEDSIP'04,Malte*, 327–331, Sep. 2004.

[Bors 98] A.G. Bors and I. Pitas. Image watermarking using block site selection and DCT domain constraints. *Optics Express*, 3(12), pp. 512–522, 1998.

[Bors 99] A. Bors I. and Pitas. Image watermarking using DCT domain constraints. In *Proceedings ICIP, vol. 3, Lausanne, Switzerland*, 231–234, 1999.

[Bouchouicha 00] M. Bouchouicha, W. Puech, A. Kolesnikov, G. Passail, and M. Dumas. Visualisation d'images hautes résolution an travers d'un arpenteur : application á l'imagerie médicale. In *CORESA'00, Poitiers, France*, 395–402, 2000.

[Caselles 93] V. Caselles, F. Catté, T. Coll, and F. Dibos. A Geometric Model for Active Contours in Image Processing. *In Numerische Mathematik*, 66, pp. pp. 1–31, 1993.

[Caselles 95] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contour. In *Proc.* 5^{th} *International Conference of Computer Vision,Boston, MA*, pp. 694–699, June 1995.

[Caselles 97] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *International Journal of Computer Vision*, 1(22), pp. pp. 61–79, 1997.

[Chakraborty 96] A. Chakraborty, L. Staib, and J. Ducan. Deformable Boundary Finding in Medical Images by Integrating Gradient and Region Information. *IEEE Transactions on Medical Imaging*, 15, pp. pp. 859–870, Dec. 1996.

[Chan 04] C.-K. Chan and L.M. Cheng. Hiding Data in Images by Simple LSB Substitution. *Pattern Recognition*, 37, pp. 469–474, 2004.

[Chang 01] C.C. Chang, M.S. Hwang, and T-S Chen. A new encryption algorithm for image cryptosystems. *The Journal of Systems and Software*, 58, pp. 83–91, 2001.

[Chang 02] C.-C. Chang, T.-S. Chen, and L.-Z. Chung. A Steganographic Method Based Upon JPEG and Quantization Table Modification. *Information Sciences, Elsevier*, 141, pp. 123–138, 2002.

[Chareyron 02] G. Chareyron and A. Tremeau. Watermarking of color Images based on a multi-layer process. In *CGIV'02, Poitiers, France*, 77–80, 2002.

[Chen 01] B. Chen and G.W. Wornell. Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. *IEEE Transactions on Information Theory*, 47(4), pp. 1423–1443, 2001.

[Cheng 00] H. Cheng and X. Li. Partial Encryption of Compressed Images and Videos. *IEEE Transactions on Signal Processing*, 48(8), pp. 2439–2451, 2000.

[Chesnaud 99] C. Chesnaud, P. Refregier, and V. Boulet. Statistical Region Snake-Based Segmentation Adapted to Different Physical Noise Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21, pp. pp. 1145–1156, Nov. 1999.

[Christopoulos 00] C. Christopoulos, J. Askelöf, and M. Larsson. Efficient Methods for Encoding Regions of Interest in the Upcoming JPEG2000 Still Image Coding Standard. *IEEE Signal Processing Letters*, 7(9), pp. 247–249, Sep. 2000.

[Chung 98] K.L. Chung and L.C. Chang. Large encrypting binary images with higher security. *Pattern Recognition Letters*, 19, pp. 461–468, 1998.

[Cocquerez 95] J.P. Cocquerez and S. Philipp. *Analyse d'images : filtrage et segmentation*. Masson, Paris, 1995.

[Cohen 91] L. Cohen. On Active Contour Models and Balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 1991.

[Cohen 92] I. Cohen, L. Cohen, and N. Ayache. Using Deformable Surface to Segment 3-D Images and Infer Differential Structures. *Computer Vision, Graphics, and Image Processing: Image Understanding*, Sep. 1992.

[Cotes 94] T. Cotes, A. Hill, C. J. Taylor, and J. Haslam. Use of Active Models for Locating Structure in Medical Images. *Image and Vision Computing*, 12, pp. pp. 355–365, 1994.

[Cox 97a] I.J. Cox, J. Killian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Trans. on Image Processing*, 6(12), pp. 1673–1687, 1997.

[Cox 97b] R.D. Cox, C.J. Henri, and P.M. Bret. CD-Based Image Archival and Management on a Hybrid Radiology Intranet. *The Canadian Journal of Medical Radiation and Technology*, 28(3), 1997.

[Cox 98] R.D. Cox, C.J. Henri, R.K. Rubin, and P.M. Bret. Dicom-Compliant PACS with CD-Based Image Archival. In *SPIE Medical Imaging, San Diego, CA*, vol. 3339, 1998.

[Debreuve 99] E. Debreuve, M. Barlaud, G. Aubert, I. Laurette, and J. Darcourt. Spice Time Segmentation Using Level Set Active Contours Applied to Myocardial Gated. In *SPECT Proceeding of Medical Imaging Conference Seattle*, 1999.

[Deguillaume 02] F. Deguillaume, S. Voloshynovskiy, and T. Pun. Hybrid robust watermarking resistant against copy attack. In *EUSIPCO'02, Toulouse, France*, 2002.

[Delaigle 98] J.F. Delaigle, C. De Vleeschouwer, and B. Macq. Watermarking Algorithm Based on a Human Visual Model. *Special Issue on Watermarking, Signal Processing*, 66(3), pp. 319–336, 1998.

[Delingette 00] H. Delingette and J. Montagnat. Topology and Shape Constraints on Parametric Active Contours. Technical report, INRIA France, January 2000.

[Diffie 76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 26(6), pp. 644–654, 1976.

[Djemal 02] K. Djemal, W. Puech, and B. Rossetto. Active Contours Propagation in a Medical Images Sequence with a Local Estimation. In *Proc.* 11^{th} *European Signal Processing Conference (EUSIPCO-2002), Toulouse, France*, Sep. 2002.

[Djemal 03a] K. Djemal, S. Paris, M. Grimaldi, W. Puech, and B. Rossetto. Réseau d'imagerie médicale et d'aide au diagnostic (RIMAD). In *Proc. 1st International Conference of Sciences of Electronic, Technology of Informations and Telecommunications, (SETIT'03), Sousse, Tunisia*, Mar. 2003.

[Djemal 03b] K. Djemal, W. Puech, and B. Rossetto. Restauration par minimisation de la variation totale adaptée à un modèle de bruit ultrasonore. In *Proc. 19th. Colloque Traitement du Signal et des Images (GRETSI'03), Paris, France*, Sep. 2003.

[Djemal 04] K. Djemal, W. Puech, and B. Rossetto. Geometric Active Contour Model Using Level Set Methods for Objects Tracking in Images Sequences. In *Proc. 2nd International Conference of Sciences of Electronic, Technology of Informations and Telecommunications, (SETIT'04), Sousse, Tunisia*, Mar. 2004.

[Djemal 05] K. Djemal, W. Puech, and B. Rossetto. Automatic Active Contours Propagation in a Sequence of Medical Images. *International Journal of Image and Graphics (IJIG)*, 5(4), 2005.

[DR02] J. Daemen and V. Rijmen. AES Proposal: The Rijndael Block Cipher. Technical report, Proton World Int.l, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, 2002.

[Droogenbroeck 02] M. Van Droogenbroeck and R. Benedett. Techniques for a Selective Encryption of Uncompressed and Compressed Images. In *Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium*, Sept. 2002.

[Ducottet 97] C. Ducottet and J. Fayolle. Détection et suivi d'interfaces d'objets déformables : application à la mécanique des fluides. In *Proc. 16th. Colloque Traitement du Signal et des Images GRETSI'97, Grenoble, France*, pp. 1487–1490, 1997.

[Duric 01] Z. Duric. *Information Hiding, Steganography and Watermarking - Attacks and Countermeasures*. Kluwer Academic Publishers, Boston, 2001.

[Fiebich 97] M. Fiebich, M.T. Mitchell, and K.R. Hoffmann. Comparison of Automatic and Manual 3D Segmentation in CT Angiography of the Abdominal Aorta. In *Scientific Program Radiology Society of North America, Chicago, Illinois*, pp. 474, 1997.

[Fillinger 99] M. Fillinger. Postoperative Imaging After Endovascular AAA Repair. *Seminars in Vascular Surgery*, 12(4), pp. pp. 327–338, 1999.

[Fish 04] M. M. Fisch, H. Stgner, and A. Uhl. Layered Encryption Techniques for DCT-Coded Visual Data. In *European Signal Processing Conference (EUSIPCO) 2004, Vienna, Austria*, Sep., 2004.

[Florescu 00] D. Florescu, V. Issarny, P. Valduriez, and K. Yagoub. Caching Strategies for Data-Intensive Web Sites. Technical Report 3871, INRIA Rocquencourt, Le Chesnay, France, January 2000.

[Fridrich 98] J. Fridrich. Applications of Data Hiding in Digital Images. In *ISPACS'98 Conference*, 1998.

[Fridrich 02a] J. Fridrich and M. Goljan. Practical steganalysis: state-of-the-art. In *Proceeding of SPIE Photonics West, Electronic Imaging 2002*, vol.

4675, 1–13, 2002.

[Fridrich 02b] J. Fridrich, M. Goljan, and R. Du. Lossless Data Embedding New Paradigm in Digital Watermarking. *EURASIP Journal Applications on Signal Processing*, 2002, pp. 185–196, Feb 2002.

[Fridrich 04] J. Fridrich, M. Goljan, Q. Chen, and V. Pathak. Lossless Data Embedding with File Size Preservation. In *Proc. EI SPIE San Jose, CA*, Jan 2004.

[Gao 97] L. Gao, D.G. Heath, and E.K. Fishman. Medical Image Segmentation Using Deformable Surface Model. In *Scientific Program Radiology Society of North America, Chicago, Illinois*, pp. 474–475, 1997.

[Gokturk 01] S.B. Gokturk, C. Tomasi, B. Girod, and C. Beaulieu. Medical image compression based on region of interest, with application to colon CT images. In *Engineering in Medicine and Biology Society (EMBS), 23rd Annual International Conference of the IEEE*, vol. 3, 2453–2456, 2001.

[Gonzales 02] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (2nd Edition)*. Pearson Education (2002), Elsevier, 2002.

[Guillem 02] S. Guillem-Lessard. http://www.uqtr.ca/∼delisle/Crypto. In ., 2002.

[Hartung 99] F. Hartung and M. Kutter. Multimedia watermarking techniques. In *IEEE Proceeding 87*, 1079–1107, 1999.

[Healey 89] G. Healey. Using Color for Geometry-Insensitive Segmentation. *Journal of the Optical Society of America-A*, 6(6), pp. 920–937, June 1989.

[Henri 97a] C.J. Henri, R.D. Cox, and P.M. Bret. Implementation of a Mini Picture Archiving and Communication System in Ultrasonography: Experience after one Year of Use. *Journal of Digital Imaging*, 10(3), pp. 80–82, 1997.

[Henri 97b] C.J. Henri, R.K. Rubin, R.D. Cox, and P.M. Bret. Design and Implementation of WWW-Based Tools For Image Management in CT, MRI and Ultrasound. *The Canadian Journal of Medical Radiation and Technology*, 28(3), pp. 135–138, 1997.

[Honsinger 01] C.W. Honsinger, P.W. Jones, M. Rabbani, and J.C. Stoffel. Lossless Recovery of an Original Image Containing Embedded Data. *US Pat. 6,278,791*, 2001.

[Huffman 62] Huffman. A Method for the Construction of Minimum Redundancy Codes. In *Proceedings IRE*, vol. 40, 1098–1101, 1962.

[HW99] C.-T. Hsu and J.-L. Wu. Hidden Digital Watermarks in Images. *IEEE Transaction on Image Processing*, 8, pp. 58–68, 1999.

[Jag] G. Jagpal. Steganography in Digital Images. In *Dissertation, University of Cambridge, Selwyn College.*

[Jayaraman 97] M. Jayaraman, B. Kimia, H. Tek, G.A. Tung, and J.M. Rogg. Semiautomated Image Segmentation of Primary Brain Tumors Based on Deformable Bubbles. In *Scientific Program Radiology Society of North America, Chicago, Illinois*, pp. 168, 1997.

[Johnson 98] N.F. Johnson and S. Jajodia. Steganalysis: The Investigation of Hidden Information. In *1998'IEEE Information Technology Conference, Syracuse, New York*, 1998.

[Kass 88] M. Kass, A. Witkins, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1988.

[Kerckhoffs 83] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, 9, pp. 5–38, 1883.

[Kervrann 94] C. Kervrann and F. Heitz. A Hierarchical Statistical Framework for the Segmentation of Deformable Objects in Image Sequence. In *In IEEE Conf. Comp. Vision Pattern Recognition, Seattle, USA*, pp. 724–728, 1994.

[Koch 95] E. Koch and J. Zhao. Towards robust and hidden image copyright labeling. In *IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, Greece*, 452–455, 1995.

[Kunkelmann 98] T. Kunkelmann. Applying Encryption to Video Communication. In *Proceedings of the Multimedia and Security Workshop at ACM Multimedia '98, Bristol, England*, 41–47, Sep. 1998.

[Kutter 98] M. Kutter. Watermarking resisting to translation, rotation and scaling. In *Proceeding of SPIE Multimedia Systems and Applications,vol 3528, Boston, USA*, 423–431, Nov 1998.

[Lacroix 00] L. Lacroix, N. Le Prince, C. Boggero, and C. Lauer. *Programmation Web avec PHP.* Eyrolles, Paris, France, 2000.

[Latombe 97] B. Latombe, P. Planet-Ladret, F. Granada, and P. Villemain. Algorithme de contour actif appliqué à la poursuite d'avalanche. In *Proc. GRETSI-97, Grenoble, France*, Sep. 1997.

[Liu 03] X. Liu and A. Eskicioglu. Selective Encryption of Multimedia Content in Distribution Networks:Challenges and New Directions. In *IASTED Communications, Internet & Information Technology (CIIT), USA*, November, 2003.

[Lovarco 03a] G. Lo-varco, W. Puech, and M. Dumas. DCT-Based Watermarking Method Using Error Correction Codes. In *ICAPR'03, International Conference on Advances in Pattern Recognition, Calcutta, India*, 347–350, 2003.

[Lovarco 03b] G. Lovarco, W. Puech, and M. Dumas. Tatouage d'images couleurs avec CCE : application à la sécurité routière. In *Proc. 8th Colloque Compression et Représentation des Signaux Audiovisuels, (CORESA'03), Lyon, France*, Jan. 2003.

[Lovarco 04a] G. Lo-Varco, W. Puech, and M. Dumas. DCT-Based Watermarking Method Using Color Components. In *Proc.* 2^{nd} *European Conference on Color in Graphics, Imaging and Vision (CGIV-04), Aachen, Germany*, 146–150, Apr. 2004.

[Lovarco 04b] G. Lovarco, W. Puech, and M. Dumas. Tatouage couleur par DCT basé sur le contenu. In *Proc. 9th Colloque Compression et Représentation des Signaux Audiovisuels, (CORESA'04), Lille, France*, 13–16, May 2004.

[Lovarco 05a] G. Lo-varco and W. Puech. DCT-Based Data-Hiding for Securing ROI of Color Images. In *International Conference on Image Processing (IEEE ICIP-2005), Genova, Italy*, Sep. 2005.

[Lovarco 05b] G. Lo-varco and W. Puech. Safe ROIs of Color Images by Inductive Data-Hiding. In *EUSIPCO'05, Antalya, Turkey*, Sep. 2005.

[Lovarco 05c] G. Lo-varco, W. Puech, and M. Dumas. Content Based Watermarking for Securing Color Images. *Journal of Imaging Science and Technology (JIST)*, 49(5), pp. 450–459, 2005.

[Malladi 95] J. Sethian R. Malladi and B.C. Vemuri. Shape Modeling With Front Propagation: a Level Set Approach. *IEEE Transactions on Pattern*

Analysis and Machine Intelligence, 17, pp. pp. 158–174, 1995.

[Maniccam 01] S.S. Maniccam and N.G. Bourbakis. Lossless image compression and encryption using SCAN. *Pattern Recognition*, 34, pp. 1229–1245, 2001.

[Maniccam 04] S.S. Maniccam and N.G. Bourbakis. Lossless Compression and Information Hiding in Images. *Pattern Recognition*, 37, pp. 475–486, 2004.

[Michelis 00] J. Michelis, W. Puech, V. Ricordel, G. Passail, and M. Dumas. Intégration d'applet JAVA dans un réseau d'images médicales : aide au télédiagnostic. In *CORESA'00, Poitiers, France*, 403–410, 2000.

[Moore 94] S.M. Moore, S.A. Hoffman, and D.E. Beecher. Dicom Shareware: A Public Implementation of the DICOM Standart. In *SPIE Medical Imaging (PACS)*, vol. 2165, 772–781, 1994.

[Mumford 89] D. Mumford and J. Shah. Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(x), pp. 577–685, 1989.

[NBS 77] NBS FIPS 46. Data Encryption Standard. Technical report, National Bureau of Standards, U.S. Department of Commerce, 1977.

[Nema 93] NEMA. Standards publication Digital Imaging and Communication in Medicine. *(DICOM)*, 1993.

[Nicolay 99] S. Nicolay, W. Puech, V. Ricordel, and G. Passail. Reconstruction 3D de l'aorte issue d'une séquence d'images tomodensitométriques : analyse et amélioration par contours actifs. In *Proc. Colloque National de Recherche en IUT, CNRIUT'99, Aix en Provence, France*, pp. 117–129, June 1999.

[Nikolaidis 98] N. Nikolaidis and I. Pitas. Robust Image Watermarking in the Spatial Domain. *Signal Processing*, 66(3), pp. 385–403, 1998.

[Norcen 03] R. Norcen, M. Podesser, A. Pommer, H.P. Schmidt, and A. Uhl. Confidential Storage and Transmission of Medical Image Data. *Computers in Biology and Medicine*, 33, pp. 277–292, 2003.

[Odobez 95a] J.M. Odobez and . Bouthemy. Robust Multiresolution Estimation of Motion Models in Complex Image Sequences. *Traitement du Signal*,

12(2), 1995.

[Odobez 95b] J.M. Odobez and P. Bouthemy. Robust Multiresolution Estimation of Motion Models. *Journal of Visual Communication and Image Representation*, 6(4), pp. pp. 348–365, 1995.

[Paragios 96] N. Paragios and R. Deriche. Geodesic Active Regions for Motion Estimation and Tracking. In *ICCV, Corfu Greece*, 1996.

[Pennebaker 93] W.B. Pennebaker and J.L. Mitchell. JPEG: Still Image Data Compression Standard. *Van Nostrand Reinhold, San Jose, USA*, 45, 1993.

[Petitcolas 99] F.A.P. Petitcolas, R. J. Anderson, and M.G. Kuhn. Information Hiding-A Survey. *IEEE, special issue on protection of multimedia content*, 87(7), pp. 1062–1078, July 1999.

[Piat 03] S. Piat. La cryptographie appliquée aux images. Technical report, Rapport de DEA Informatique, Réseaux et Image, Université de Reims, 2003.

[Piva 97] A. Piva, M. Barni, F. Bartolini, and V. Capellini. DCT Based Watermark Recovering Without Resorting to the Uncorrupted Original Image. In *International Conference on Image Processing, Austin, Texas*, 520–523, 1997.

[Puech 99] W. Puech, G. Passail, S. Nicolay, and V. Ricordel. Analyse et amélioration de méthodes de reconstruction 3D de l'aorte à partir d'une séquence d'images tomodensitométriques. In *Proc. 17th. Colloque Traitement du Signal et des Images GRETSI'99, Vannes, France*, vol. 4, pp. 1053–1056, Sep. 1999.

[Puech 00] W. Puech, G. Passail, and V. Ricordel. Analysis and Optimisation of 3D Reconstruction Method of the Aorta from a Tomographic Images Sequence. In *Proc. 10th European Signal Processing Conference, EUSIPCO'2000 Tampere, Finlande*, vol. 1, Sep. 2000.

[Puech 01a] W. Puech, J.J. Charre, and M. Dumas. Transfert sécurisé d'images par chiffrement de Vigenère. In *NimesTic 2001, La relation Homme - Système : Complexe, Nîmes, France*, 167–171, Dec. 2001.

[Puech 01b] W. Puech and M. Dumas. Transfert sécurisé d'images par combinaison de techniques de cryptographie et de tatouage. In *Proc. 7th*

Colloque Compression et Représentation des Signaux Audiovisuels, CORESA'01, Dijon, France, Nov. 2001.

[Puech 01c] W. Puech, M. Dumas, J.C.Borie, and M. Puech. Tatouage d'images cryptées pour l'aide au Télédiagnostic. In *Proc. 18th. Colloque Traitement du Signal et des Images, GRETSI'01, Toulouse, France*, Sep. 2001.

[Puech 01d] W. Puech, M. Puech, and M. Dumas. Accès sécurisé à distance d'images médicales haute résolution. In *Proc. 11th. Forum des Jeunes Chercheurs en Génie Biologique et Médical, Compiègne, France*, 72–73, Jun. 2001.

[Puech 02] W. Puech, P. Montesinos, and M. Dumas. Color Image Watermarking Robust to JPEG Compression. In *Proc.* 1^{st} *European Conference on Color in Graphics, Imaging and Vision (CGIV-02), Poitiers, France*, 81–85, Apr. 2002.

[Puech 03] W. Puech. Safe Transfer of Image Based on Color Transformation for Watermarking. In *Proc. Workshop Transmitting, Processing and Watermarking Multimedia Content, (WTPWMC'03), Bordeaux, France*, 1–6, Mar. 2003.

[Puech 04a] W. Puech and J.M. Rodrigues. A New Crypto-Watermarking Method for Medical Images Safe Transfer. In *EUSIPCO'04, Vienna, Austria*, 2004.

[Puech 04b] W. Puech and J.M. Rodrigues. Sécurisation d'image par crypto-tatouage. In *Proc. 9th Colloque Compression et Représentation des Signaux Audiovisuels, (CORESA'04), Lille, France*, 215–218, May 2004.

[Puech 05] W. Puech and J.M. Rodrigues. Crypto-Compression of Medical Images by Selective Encryption of DCT. In *EUSIPCO'05, Antalya, Turkey*, Sep. 2005.

[Puech 06] W. Puech and J.M. Rodrigues. An Autonomous Crypto-Data Hiding Method for Images Safe Transfer. *Signal Processing: Image Communication (SPIC)*, 2006.

[Qiao 98] L. Qiao and K. Nahrstedt. Comparison of MPEG Encryption Algorithms. *International Journal on Computers and Graphics (Special*

Issue on Data Security in Image Communication and Networks, 22(3), pp. 437–444, 1998.

[Queluz 99] M.P. Queluz. Content-based integrity protection of digital images. In *SPIE Proceeding 3657*, 85–93, 1999.

[Ricordel 99] V. Ricordel, W. Puech, G. Passail, and A. Kolesnikov. Conception d'un réseau pour la communication d'images médicales. In *Proc. 5th Conference Compression et Représentation des Signaux Audiovisuels, CORESA'99, Sophia-Antipolis, France*, 229–236, june 1999.

[Rivest 78] A. Shamir R. L. Rivest and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), pp. 120–126, 1978.

[Rodrigues 04a] J.M. Rodrigues, W. Puech, and C. Fiorio. Lossless Crypto-Data Hiding in Medical Images Without Increasing the Original Size. In *Proc. 2nd International Conference on Advances in Medical Signal and Information Processing, MEDSIP'04,Malte*, 358–365, Sep. 2004.

[Rodrigues 04b] J.M. Rodrigues, J.R. Rios, and W. Puech. SSB-4 System of Steganography using Bit. In *5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2004), Lisboa, Portugal*, april 2004.

[Rodrigues 06] J.M. Rodrigues and W. Puech. An Adaptable Invertible Crypto-Data Hiding Method for Still Heterogeneous Images. *EURASIP Journal on Applied Signal Processing*, 2006.

[Rosenfeld 68] A. Rosenfeld and J. Pfaltz. Distance Functions in Digital Pictures. *Pattern Recognition*, 1, pp. 33–61, June. 1968.

[Ruanaidh 96] Ruanaidh J.J.K., Dowling W.J., and Boland F.M. Phase Watermarking of Digital Images. In *International Conference on Image Processing, Lausanne, Switzerland*, 1996.

[Ruanaidh 98] J.J.K. O Ruanaidh and T. Pun. Rotation, scale and translation invariant spread spectrum digital image watermarking. In *Signal Processing 66*, 303–317, 1998.

[Santosa 96] F. Santosa. A Level Set Approach for Inverse Problems Involving Obstacles. *ESAIM*, 1996.

[Schneier 95] B. Schneier. *Applied cryptography.* Wiley, 1995.

[Schneier 97] B. Schneier. *Cryptographie appliquée : protocoles, algorithmes et codes sources en C*. Wiley, 1997.

[Sclaroff 99] S. Sclaroff, M. La Caxias, S. Sethi, and L. Taycher. Unifying Textual and Visual Cues for Content Image Retrevial on the World Wide Web. *Computer Vision and Image Understanding*, 75, pp. 86–98, 1999.

[Sec03] RSA Security. http://www.rsasecurity.com/. In ., 2003.

[Serra 88] J. Serra. *Image Analysis and Mathematical Morphology, vol. 2*, vol. 2. London: Academic Press, 1988.

[Sethian 96] J.A. Sethian. Level Set Methods. *Cambridge University Press*, 1996.

[Shih 03] F. Y. Shih and S. Y.T. Wu. Combinational image watermarking in the spatial and frequency domains. *Pattern Recognition*, 36, pp. 969–975, 2003.

[Silman 01] J. Silman. Steganography and Steganalysis: An Overview. In *Sans InfoSec Reading Room*, 2001.

[Sinha 03] A. Sinha and K. Singh. A technique for image encryption using digital signature. *Optics Communications*, 218, pp. 229–234, 2003.

[Smyth 96] P.P. Smyth, C.J Taylor, and J.E. Adams. Automatic Measurement of Vertebral Shape using Active Shape Models. In *In Proc. 3rd. IEEE Worshop on Applications of Computer Vision, Sarasota, Florida, USA*, pp. 176–180, 1996.

[Stinson 95] D. Stinson. *Cryptography - Theory and Practice*. CRC Press, 1995.

[Stinson 96] D. Stinson. *Cryptographie - Théorie et pratique*. Thompson Publishing, 1996.

[Strom 97] J. Strom and P. Cosman. Medical Image Compression With Lossless Regions of Interest. *Signal Processing*, 59(2), pp. 155–171, June 1997.

[Tabaty 00] M-L. Tabaty, N. Ayache, J. Darcourt, and G. Malandin. Analyse statistique d'images médicales : étude et utilisation du logiciel SPM. Technical Report RR-3802, INRIA, Sophia-Antipolis, France, 2000.

[Tang 96] L. Tang. Methods for Encrypting and Decrypting MPEG Video Data Efficiently. In *ACM Multimedia*, 219–229, 1996.

[Tanguy 97] J-P. Tanguy. Application du lidar à la détection d'objets flottants ou faiblement immergés. In *Proc. 16th. Colloque Traitement du Signal et des Images GRETSI'97, Grenoble, France*, pp. 741–744, 1997.

[Toutant 05a] J.L. Toutant, W. Puech, and C. Fiorio. Asynchronous DCT-Based Data-Hiding Robust to Cropping. In *6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005), Montreux, Switzerland*, april 2005.

[Toutant 05b] J.L. Toutant, W. Puech, and C. Fiorio. Amélioration de l'invisibilité par adaptation de la quantification aux données à insérer. In *20th. Colloque Traitement du Signal et des Images (GRETSI'05), Louvain-la-Neuve, Belgique*, september 2005.

[Tseng 04] H.-W. Tseng and C.-C. Chang. High Capacity Data Hiding in JPEG-Compressed Images. *Informatic, Institute of Mathematics and Informatic, Vilnius*, 151(1), pp. 127–142, 2004.

[Udupa 97] J.K. Udupa. 3D Imaging Methodologies : A Current Perspective. In *Scientific Program Radiology Society of North America, Chicago, Illinois*, pp. 625, 1997.

[Upham 97] D. Upham. Jpeg-jsteg, Modification of the Independent Jpeg Group's Jpeg Software for 1-bit Steganography in Jfif Output Files. In *ftp://ftp.funet.fi/pub/crypt/steganography*, 1997.

[Voyatzis 99] G. Voyatzis and I. Pitas. The use of watermarks in the protection of digital multimedia products. In *IEEE Proceeding 87*, 1197–1207, 1999.

[Wakatani 02] A. Wakatani. Digital Watermarking for ROI Medical Images by Using Compressed Signature Image. In *35th Hawaii International Conference on System Sciences*, 2002.

[Wallace 91] G. Wallace. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34(4), pp. 31–44, April 1991.

[Wen 02] J. Wen, M. Severa, W. Zeng, M. Luttrell, and W. Jin. A Format-Compliant Configurable Encryption Framework for Access Control of Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6), pp. 545–557, 2002.

[Wenjing 04] W. Jia, X. He, and Q. Lin. Echocardiography Sequential Images Compression Based on Region of Interest. In *2nd International Conference on Information Technology for Application,ICITA'04*, 2004.

[Wheeler 94] D. Wheeler and R. Needham. http://www.ftp.cl.cam.ac.uk/ftp/papers/djw-rmn/djw-rmn-tea.html. In ., 1994.

[Winkler 99] S. Winkler and M. Kutter. Vers un tatouage à étalement de spectre optimal utilisant le système visuel humain. In *Proc. 5th Colloque Compression et Représentation des Signaux Audiovisuels, (CORESA'99), Sophia Antipolis, France*, 1999.

[Wu 96] X. Wu and N. Memon. CALIC - Context Based Adaptive Lossless Image Codec. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 4, pp. 1890–1893, May 1996.

[Yhang 00] J. Yhang, H. Choi, and T. Kim. Noise Estimation for Blocking Artifacts Reduction in DCT Coded Images. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(7), 2000.

[Zeng 99] W. Zeng and S. Lei. Efficient Frequency Domain Video Scrambling for Content Access Control. In *ACM Multimedia, Orlando, FL, USA*, 285–293, Nov. 1999.

[Zhu 95] S.C. Zhu, A. Yuille, and T. S. Lee. Region Competition: Unifying Snakes, Region Growing and Bayes/MDL for Multiband Image Segmentation. In *in Proc. Int. Conf. Computer Vision*, pp. 416–423, 1995.

[Zhu 96] S. Zhu and A. Yuille. Region Competition: Unifying Snakes, Region Growing and Bayes/MDL for Multiband Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18, pp. 884–900, 1996.