INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Une introduction à la compression d'images médicales volumiques

Jonathan Taquet — Claude Labit

N° 7324

Juin 2010

ISSN 0249-6399 ISRN INRIA/RR--7324--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
RENNES - BRETAGNE ATLANTIQUE

Une introduction à la compression d'images médicales volumiques

Jonathan Taquet*, Claude Labit†

Thème : Compression sans perte et presque sans perte
Équipe-Projet Temics

Rapport de recherche n° 7324 — Juin 2010 — 93 pages

Résumé : Ce rapport est une introduction à la compression d'image médicales volumiques. Il en présente le contexte de recherche, dresse un état de l'art des techniques existantes, et commente les résultats de quelques expérimentations.

Nous présentons deux modalités d'acquisition produisant des images médicales volumiques et utilisées de manière intensive : l'imagerie par résonance magnétique et la tomographie. Les longues périodes d'archivage et la consultation de ces images volumineuses au travers de réseaux nécessitent des algorithmes de compression performants et offrant des facilités de navigation. Afin de satisfaire l'éthique des médecins, la compression doit généralement être effectuée sans détérioration de l'image originale. L'état de l'art se focalise donc sur les techniques n'introduisant aucune perte. Les algorithmes concernant des images bi-dimensionnelles approchent les performances théoriques optimales. Bien que souvent plus performantes, les extensions volumiques, n'offrent que de faibles taux de compression (ces taux peuvent être inférieurs à 2:1 et dépassent rarement 6:1). Afin d'être plus efficace il faut inévitablement introduire des pertes d'information, l'étude s'est donc également intéressée aux techniques de compression introduisant des pertes contrôlées, à savoir la compression de régions d'intérêt et la compression presque sans perte.

Nous comparons les résultats de la compression bi et tri-dimensionnelle et montrons que les gains d'une approche volumique varient en fonction de la qualité de l'image (résolution, algorithme de construction de l'image, bruit). Puis nous commentons les améliorations potentielles de l'utilisation de régions d'intérêt ou d'algorithmes presque sans perte, qui peuvent permettre de réduire la taille des fichiers compressés de 20% à plus de 60% avec une perte d'information raisonnable.

Mots-clés : images médicales volumiques, IRM, tomographie, compression, sans perte, presque sans perte, région d'intérêt

Ce travail est supporté par le contrat de recherche doctoral INRIA n°4591 et cofinancé par la région Bretagne.

* Doctorant, Jonathan.Taquet@inria.fr
† Directeur de recherches, Claude.Labit@inria.fr

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

An Introduction to the Compression of Volumetric Biomedical Images

Abstract: This report is an introduction to the compression of volumetric biomedical images. It introduces the biomedical context, surveys the state of the art and comments some experimental results.

We present two intensively used biomedical acquisition modalities, the Magnetic Resonance Imaging and the Computed Tomography that produce volumetric images. Due to the amount of data, the wide archival period, and the remote consultation through networks which is a common practice, this biomedical images needs powerfull compression algorithms that allow navigation facilities. But to satisfy the ethics of radiologist the compression must often be done without any degradation. The state of the art is therefore focused on lossless image compression. The current 2D lossless algorithms propose near optimal compression results and their 3D extensions, even if generaly better, provide poor results (compression rates can be less than 2:1 and hardly better than 6:1). Since some loss can not be avoided to be more efficient, the study has also focused on controlled loss : region of interest based compression, and near-lossless compression.

We compare the results of the 2D and 3D compression and we show that the gains induced by volumic approach can vary with the image quality (resolution, algorithms used for the computation of the image, noise). Then we comment the expected compression improvements with the use of région of interest or near-lossless algorithms : with reasonable losses, the size of compressed files can be reduced by 20% to 60%.

Key-words: volumetric biomedical images, MRI, CT, compression, lossless, near lossless, region of interest

Table des matières

Chapitre 1

Introduction

Ces dernières décennies, l'imagerie médicale connaît une évolution spectaculaire aussi bien au niveau du développement des techniques de production qu'au niveau de leur utilisation. Aujourd'hui, ces nouvelles technologies se rendent indispensables pour les diagnostics et leur usage intensif pose des problèmes de stockage et de transmission.

Ce chapitre introduira le concept d'image médicale volumique, le vocabulaire associé, ainsi que les modalités d'images et les techniques d'acquisitions concernées par ce document. Le contexte dans lequel s'insère ce travail sera ensuite décrit afin de mettre en évidence la nécessité d'une compression efficace et adaptée à l'utilisation de ces images. Suivront une mise en garde sur les difficultés pouvant survenir lors de l'utilisation de ces clichés numériques médicaux, et un descriptif des bases d'images utilisées lors de cette étude. Enfin, l'organisation de ce document sera détaillée.

Pour toute information complémentaire à cette introduction, le lecteur pourra également se référer aux chapitres 1 et 3 de [NACM07, NACM08], respectivement consacrés à l'intérêt de la compression en milieu médical et à la présentation de diverses modalités d'acquisition.

1.1 Images médicales volumiques

Hormis la radiographie, les technologies les plus fréquemment utilisées dans le domaine médical permettent d'acquérir des images volumiques. Ces volumes sont organisés comme une succession d'images bidimensionnelles (coupes) prises à des distances régulières le long d'un 3ème axe transversal. Le pixel d'une image appartenant à une telle séquence ne correspond plus à l'intégration bidimensionnelle d'une énergie projetée sur un capteur 2D, mais à la quantification d'une énergie présente dans un petit volume. C'est pourquoi il est couramment appelé voxel.

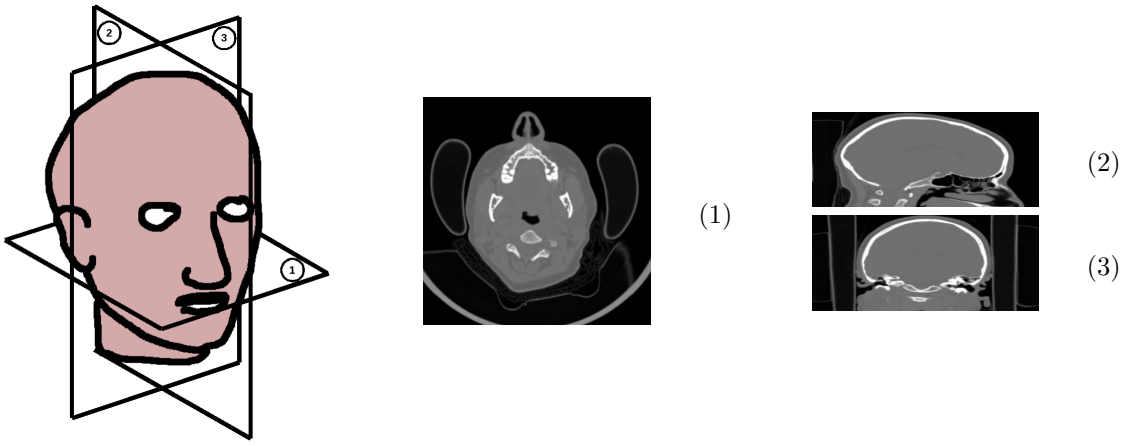


Fig. 1.1 – Représentations : transverse (1), sagittale (2), coronale (3)

La visualisation coupe à coupe du volume peut se faire le long de chacun des 3 axes d'acquisition. Le plus souvent les coupes sont successivement obtenues transversalement à l'axe tête/pieds du patient qui est allongé sur une table. Une telle organisation des images est alors dite « représentation transverse » ou « transaxiale ». Une « représentation sagittale » correspond à une organisation des voxels pour former des coupes perpendiculaires à l'axe épaule/épaule. Enfin une « représentation coronale » est triée le long de l'axe dos/ventre (cf. figure 1.1).

En se plaçant dans le repère d'acquisition original, la résolution spatiale inter-coupe (axe z) n'est pas nécessairement la même que la résolution spatiale interne à la coupe (axes x,y). Ainsi, le volume d'un voxel n'est pas toujours cubique. Il peut également y avoir un recouvrement entre coupes successives, lorsque leur épaisseur est plus importante que le pas d'acquisition.

Dans les sections suivantes seront présentées les deux modalités d'images volumiques les plus répandus dans le domaine médical, et autour desquelles ce travail s'est orienté : la TomoDensitoMétrie (TDM), dite aussi tomographie axiale calculée par ordinateur, CT-scan (*Computed Tomography*) ou plus simplement scanner ; et l'Imagerie par Résonance Magnétique (IRM).

1.1.1 La TomoDensitoMétrie (TDM)

L'acquisition d'une coupe tomographique se fait à l'aide de multiples faisceaux de rayons X. Ils sont projetés selon plusieurs orientations le long du plan à observer et ceux ayant réussi à traverser l'objet (individu) à analyser (selon ce plan) sont captés pour chacune des orientations. Les énergies résultantes de ces projections sont quantifiées et numérisées afin d'être utilisée pour calculer une image.

Plusieurs générations de matériels d'acquisition ont vu le jour : les plus anciens prenaient les coupes une à une, la table avançant par pas et s'arrêtant pour chaque prise. Dans les années 90 un tel examen pouvait prendre un peu moins d'une heure, et des artefacts importants dus aux mouvements (respiration par exemple) du patient pouvaient apparaître. Cette technique a été quasiment abandonnée. Le scanner hélicoïdale, quant à lui, fait avancer la table de manière constante, les faisceaux de rayons X décrivent alors une hélice autour du patient et l'acquisition se fait de façon continue. Les coupes acquises ainsi sont souvent assez épaisses et les représentations coronales et sagittales souffrent d'un manque de précision. Enfin, les techniques actuelles permettent de prendre jusqu'à 128 (et plus) coupes simultanément à l'aide de plusieurs rangées de détecteurs (scanners multibarrettes). Le tube à rayons X effectue une rotation complète autour du patient en un temps inférieur à la demi seconde pour le matériel le plus récent.

L'épaisseur des coupes peut varier entre 0.5mm et 5mm et atteindre 10mm sur un matériel plus ancien. Certains scanners permettent également de choisir parmi un gamme d'épaisseurs, ainsi qu'entre une acquisition axiale ou hélicoïdale.

Un exemple de matériel récent, inauguré fin 2008 au centre hospitalier Duchenne à Boulogne, est le *Somaton Définition AS+* de la société Siemens qui comporte 128 barrettes et permet une acquisition d'un corps entier en 4 secondes. Il effectue une rotation complète en 0.3 secondes, a une résolution spatiale de 0.33mm, une épaisseur de coupe de 5mm et une distance inter-coupe de 3mm.

Les faisceaux de rayons X projetés selon diverses orientations permettent d'obtenir le *sinogramme des projections*. Cette représentation est similaire à l'espace des projections de Radon :

$$p(u,\theta) = \int_{-\infty}^{\infty} f(t\sin\theta + u\cos\theta, -t\cos\theta + u\sin\theta)\, dt,$$

et son inverse :

$$f(x,y) = \int_{0}^{\pi} p(x\cos\theta + y\sin\theta, \theta) d\theta.$$

La discrétisation de u et θ, pouvant varier selon le matériel, compromet la reconstruction parfaite de l'image. Un certain nombre de traitements doivent alors être effectués afin d'améliorer le rendu.

La méthode d'inversion la plus répandue dans les dispositifs d'acquisition est la rétroprojection filtrée FBP (*Filtered BackProjection*) : les coefficients de chacune des orientations θ sont tout d'abord filtrés avant d'être rétroprojetés à l'aide d'un algorithme itératif. Cette technique de reconstruction repose sur le théorème de la coupe centrale CST (*Central Slice Theorem*) qui met en évidence la relation entre la transformée de Fourier de $p(u,\theta)$ selon u, et la transformée de Fourier de l'image dans un espace polaire. Ainsi, l'espace de Fourier de l'image acquise à l'aide d'un sinogramme est discrétisée selon une grille polaire et l'énergie se trouve concentrée autour des basses fréquences. De façon théorique (lorsque le pas d'échantillonnage tend vers 0), pour que la rétroprojection soit une inversion exacte de la transformée,

la représentation dans l'espace de Radon doit d'abord être filtrée par un filtre rampe (qui permet de normaliser l'énergie).

Dans la pratique, la représentation discrétisée filtrée permet seulement de reconstruire un approximation de l'image originale et génère des artefacts. Ces artefacts, bien que réduits par le filtrage, sont tout de même visibles : des halos de formes étoilées ou des droites diffusant l'énergie des zones ou celle-ci est concentrée de manière importante peuvent apparaître. Il est à noter que l'acquisition des sinogrammes souffre (comme en radiographie) d'un bruit poissonnien qui perturbe également la reconstruction. Afin de réduire les artefacts, il est possible d'utiliser d'autres filtres que le filtre rampe. Cependant, l'image générée aura tendance à subir d'autres distorsions (être plus floue par exemple). Bertram et al. [BRS+04] proposent également un algorithme d'interpolation du sinogramme avant la reconstruction afin d'augmenter la précision dans l'espace de Radon. Des techniques similaires devraient également permettre de générer des images volumiques avec une résolution plus importante que celle d'acquisition.

Il existe diverses autres techniques d'inversion du sinogramme qui ne provoquent pas forcément les mêmes artefacts. Elle peuvent être catégorisées en deux familles : les méthodes algébriques qui résolvent un système d'équations linéaires (ART, SIRT, ILST, méthode du gradient conjugué, etc.) [GBH70, Gil72, HR73, HLL75] et qui ignorent la présence du bruit dans les données, contrairement aux méthodes statistiques (MLEM, OSEM, RAMLA, DRAMA, etc.) [SV82, LBL87, HL94, BdP96, TK03].

Des techniques de reconstruction ont été étendues pour utiliser l'information tridimensionnelle (sinogrammes de plusieurs coupes) et prendre en considération le positionnement (géométrie) des capteurs. Ceci permet de réduire le bruit ainsi que les artefacts de reconstruction. Parmi les approches volumiques, on peut citer la généralisation de la rétroprojection filtrée (3D FBP).

1.1.2 l'Imagerie par Résonance Magnétique (IRM)

L'IRM utilise les propriétés magnétiques quantiques (spin) de certains atomes comme l'hydrogène qui est très présent dans les molécules composant les tissus biologiques, telles que l'eau. Des ondes magnétiques oscillantes sont appliquées à ces atomes, qui, déjà soumis à un fort champ magnétique constant $\vec{B}$ vont entrer en résonance magnétique (écho de spin). Ce phénomène survient à une fréquence particulière ω : la fréquence de Larmor ou fréquence de résonance, proportionnelle à B (et différente selon les atomes), et se traduit par une modification de l'aimantation du noyau des atomes (protons). Celle-ci va effectuer un mouvement de rotation perpendiculaire au champ $\vec{B}$ avec une fréquence ω : la précession. Après l'arrêt de la fréquence excitatrice, l'aimantation continue d'osciller et va progressivement se restabiliser pour suivre le champ B. La résonance magnétique émise durant cette phase de stabilisation est captée à l'aide d'antennes et génère un signal électrique d'intensité proportionnelle à la quantité de protons en résonance qui oscille donc également à la fréquence ω. Pour pouvoir mesurer localement la concentration d'atomes, le champ magnétique $\vec{B}$ est appliqué de manière variable dans l'espace (gradient), afin de générer une fréquence de résonance différente en chacun des points à étudier. Pour obtenir des images avec un meilleur contraste, on mesure généralement le temps nécessaire aux atomes pour revenir à l'équilibre longitudinalement (relaxation T1) et transversalement (relaxation T2). Les fréquences ainsi captées permettent de former une image dans un espace couramment appelé *espace K*. Cette représentation offre un codage de la phase et de l'amplitude pour diverses fréquences et orientations et est organisée de telle sorte qu'elle forme un espace de Fourier discrétisé. Elle est ainsi facilement inversible à l'aide d'algorithmes de transformée de Fourier rapides et permet de générer l'image de la localisation des sources de résonance.

Sur un principe similaire à l'écho de spin, on trouve également l'écho de gradient, qui consiste aussi à donner un mouvement de rotation à l'aimantation du noyau, mais avec un angle plus faible ($< 90°$, $90°$ correspondant à l'écho de spin), qui permet un temps d'excitation également plus faible et une stabilisation plus rapide après arrêt de l'excitation. Ainsi, l'écho de gradient offre un gain de temps qui permet de remplir plus rapidement un espace K tridimensionnel, offrant ainsi la possibilité de construire des IRMs de façon volumique directement (IRM3D). Les coupes d'un tel volume peuvent être plus fines car leur rapport signal à bruit est meilleur, et sont reconstruites à l'aide d'une transformée de Fourier 3D inverse.

Plus le champ magnétique constant est élevé, plus le rapport signal sur bruit tend à être meilleur. Pour l'imagerie médicale, il est souvent compris entre 0.1 et 3 Tesla et peut parfois dépasser 11 Tesla. On distingue trois types d'aimants : permanents, résistifs et supraconducteurs. Les premiers produisent un champ magnétique permanent sans consommer d'énergie. Il sont donc très fiables et tendent à se développer. Ils sont cependant très lourd (plusieurs tonnes) et dépassent rarement 0.4 Tesla. L'aimant résistif se compose d'une simple bobine de cuivre qui génère un champ magnétique lorsqu'elle est traversée par un courant électrique. Bien que peu coûteux à la fabrication, il consomme beaucoup d'énergie, souffre

de l'effet Joule et génère un champ magnétique peu stable qui atteint difficilement 0.5 Tesla. C'est pourquoi il est peu utilisé depuis l'apparition des aimants supraconducteurs. Ces derniers se composent d'une bobine en Niobium-Titane (Nb-Ti) baignée constamment dans de l'hélium liquide afin de réduire sa résistivité. Ils permettent ainsi de générer de forts champs magnétiques. Ces appareils sont très coûteux à l'achat comme à l'utilisation à cause de leur consommation importante en électricité et en hélium liquide.

Il existe deux catégories de systèmes d'acquisitions : l'une dite fermée et l'autre dite ouverte. Les systèmes fermés sont les plus répandus. Ils se présentent sous la forme d'un tunnel autour duquel on trouve l'aimant, les bobines de gradient de champ magnétique et les antennes émettrice/réceptrices générant ou captant les fréquences de résonance. Les systèmes ouverts ont des configuration variables en fonction de la forme de l'aimant utilisé (« fer à cheval » par exemple). Ces derniers génèrent des champs magnétiques moins importants, mais permettent plus facilement de faire passer des IRM à des personnes claustrophobes, enceintes ou encore obèses. Les derniers modèles arrivent tout de même à dépasser les 1.0 Tesla.

Les techniques d'acquisition les plus rapides permettent de remplir l'espace K en moins d'une seconde mais sont moins précises (plus floues et plus bruitées). Ainsi la durée d'acquisition pour un examen est en général d'une dizaine de minutes. On peut actuellement obtenir des résolutions proches du demi-millimètre pour des épaisseurs de coupes pouvant varier entre 2 et 5 mm. Ces coupes épaisses sont nécessaires pour obtenir un rapport signal à bruit intéressant.

La majorité des artefacts visibles sont liés aux propriétés physiques de la technique d'acquisition, et non à la méthode de reconstruction. La présence d'autres atomes entrant en résonance ou l'orientation des tissus selon un certain angle (« *magic angle* ») génèrent une énergie qui peut perturber l'interprétation des images. De faibles mouvements du patient durant l'acquisition génèrent également des artefacts. Enfin, des oscillations peuvent apparaître aux abords des contours très marqués ; c'est le phénomène de Gibbs.

Pour obtenir des informations plus détaillées, un cours interactif très bien conçu, et récompensé par plusieurs organismes de radiologie, est disponible en anglais [HMa] et en français [HMb].

1.2 Contexte médical et légal

1.2.1 Volumes de données

Pour des raisons médicales (suivi des patients), juridiques (expertises en cas de litige) et afin d'éviter des examens redondants, les clichés médicaux doivent être archivés à plus ou moins long terme. La loi française prévoit une conservation des données relatives à un patient durant au minimum 20 ans après son dernier contact hospitalier. En cas de procès les informations doivent rester disponibles jusqu'au règlement définitif du dossier. Enfin, pour les problèmes de nature héréditaire, la conservation est illimitée dans le temps.

Ces longues périodes de conservation ajoutées à une constante évolution de l'imagerie médicale posent de sérieux problèmes d'archivage et de transfert. A titre d'exemple, la tomographie est devenue très populaire durant les dernières décennies et son usage s'est beaucoup intensifié. La quantité d'images ainsi produites chaque année a explosé de manière quasi exponentielle. Les évolutions technologiques ont également conduit à l'augmentation de la résolution (x,y,z) et de la précision d'acquisition (bits par pixels) des appareils. De ce fait, les images deviennent de plus en plus volumineuses.

L'IRM, quant à elle, offre souvent une résolution transversale (z) plus faible que la tomographie et son coût et sa durée d'acquisition rendent sa fréquence d'utilisation moins élevée. Bien qu'elle soit ainsi moins gourmande en espace de stockage, elle reste la troisième modalité la plus encombrante après les radiographies et les scanners d'après des études locales, menées au sein de centres d'archivage communément appelés PACS (*Picture Archiving and Communication System*) [LLK+05, OBO05, OBBO06].

Les coupes des IRMs ont souvent une résolution de 256×256 pixels qui tend à aller vers 512×512 sur du matériel récent et en IRM3D. 512×512 est également la résolution la plus courante pour les scanners. Lorsque les coupes ne sont pas compressées, chaque pixel est stocké sur 2 octets (16 bits dont l'utilisation effective varie entre 12 et 16 selon le matériel). En moyenne, un volume acquis sur un scanner comporte environ 300 coupes et seulement 40 pour une IRM. Cependant en IRM plusieurs volumes peuvent être acquis lors d'un examen (selon des orientations différentes, mesurant le temps de relaxation T1 ou T2, et/ou IRM3D plus précises), ce qui tend à produire 200 coupes par examen.

L'étude menée dans le PACS du département de radiologie du centre médical universitaire de Groningen aux Pays-Bas [OBO05] montre que la production des scanners a évolué de 19875 coupes par mois en

moyenne pour l'année 2000 à 552773 pour l'année 2004, tandis que celle des IRMs s'est plus faiblement accrue de 66315 à 104457. Ainsi en 2004 plus de 3 TB ont été produits par les scanners et 657 GB par les IRMs, pour un total de plus de 5 TB produits, toutes modalités d'images confondues. En Janvier 2006, leurs prévisions de production étaient de quasiment 16 TB pour l'année 2010 alors qu'elles n'étaient que de 13 TB 7 mois plus tôt [OBBO06].

Les réseaux des hôpitaux souffrent également d'une consultation accrue de ces images volumineuses. Bien qu'il existe une augmentation perpétuelle des capacités de stockage et des débits grâce aux avancées scientifiques, les coûts matériels pour le transfert et l'archivage deviennent faramineux. La compression des images volumiques en vue du stockage et/ou d'une transmission efficace est donc un enjeu important.

1.2.2 Contraintes de qualité

Stockage sans perte

Le plus souvent, pour satisfaire l'éthique des médecins, les données des images médicales sont archivées soit de manière brute (sans compression) soit après une compression sans perte. Ceci permet de conserver une copie de l'image identique à l'originale.

Le stockage brute consiste simplement à enregistrer les valeurs des voxels un à un, le plus souvent ordonnés par leurs numéros de coupe, ligne et colonne. Ce mode de stockage utilise un nombre de bits par voxel multiple de 8 (octet) de manière à faciliter les manipulations. Ainsi des scanners en niveaux de gris sur 12 bits seront stockés sur 16 bits (2 octets) par voxel, ce qui rend leur archivage encore plus coûteux.

La compression sans perte, quant à elle, peut être effectuée à l'aide d'algorithmes de compression de données généralistes, non spécifiques à l'image. Les formats tels que ZIP, RAR, GZ, BZ2, 7Z ou encore PAQ8 sont communs et aisés à mettre en place puisqu'ils permettent de s'astreindre facilement des diverses organisations numériques (données sur plus de 8 bits : little endian, big endian ; formats de fichiers ; ...). Cependant, les algorithmes les plus performants, tels que PAQ8 ou LZMA (7Z) sont également les plus complexes et restent moins efficaces que des algorithmes spécifiques à l'image. Leurs taux de compression sont souvent moins bons pour des temps de calcul pouvant être beaucoup plus importants. Même si ces techniques permettent d'obtenir un gain d'espace de stockage, elles sont moins adaptées et les algorithmes de compression d'images sans perte leurs sont donc bien souvent préférés. Cependant, bien que plus performants, ces systèmes offrent des taux de compression pouvant être inférieurs à 2:1 et dépassant rarement 6:1, aussi bien pour les algorithmes 2D que pour les algorithmes 3D.

Stockage avec pertes

Les performances de la compression sans perte sont très faibles en comparaison à des systèmes supprimant une partie de l'information et dits avec pertes. Cependant, afin d'éviter tout litige (destruction de preuves par la compression) au cours d'une expertise en cas de poursuite judiciaire, les médecins devraient effectuer leurs diagnostiques sur les mêmes images que celles archivées. Ainsi, si une image doit être compressée avec pertes, celle présentée au radiologue devrait être identique et donc posséder les mêmes dégradations. Les médecins sont très réticents à une telle mesure. Celle-ci impliquerait la possibilité qu'une information importante pour le diagnostique en cours (ou un diagnostique ultérieur) soit altérée et introduise des faux-négatif ou faux-positifs, ou rende un cliché non interprétable. En effet, l'information utile au diagnostique est parfois de l'ordre de quelques pixels (cf. [NACM07, NACM08] chapitre 5). C'est pourquoi la majorité des travaux en compression d'images médicales s'attachent à conserver la totalité de l'information (compression sans perte).

Bien que des erreurs de diagnostique puissent être introduites par les pertes liées à la compression, elles peuvent également être causées par le bruit et les artefacts de construction spécifiques au système d'acquisition, ou tout simplement par des erreurs humaines. Ces erreurs de reconstruction liées aux contraintes physiques du matériel étant inévitables, les médecins se sont habitués à travailler avec elles, et les artefacts leurs sont devenus familiers. Ainsi, si un processus de compression avec pertes devait être instauré, il devrait être acceptable d'obtenir des taux d'erreurs de diagnostique similaires à ceux actuels après un apprentissage et une adaptation des radiologues a ces nouvelles perturbations.

La compression avec pertes des images médicales peut également chercher à conserver l'information nécessaire à ne pas perturber le diagnostique habituel. L'approche la plus satisfaisante, en terme de qualité, consiste à compresser sans perte la ou les régions d'intérêt diagnostique (ROI : *Region Of Interest*) et à

compresser avec pertes les zones pouvant être considérées comme d'un intérêt diagnostique nul (RONI : *Region Of No Interest*). Le plus souvent la ROI se compose de la totalité des pixels appartenant au patient, et la RONI des pixels extérieurs (les autres). Les contraintes de cette approche sont la nécessité de devoir définir la zone d'intérêt (effectué manuellement par le radiologue ou algorithmiquement) pour un gain de compression, certes moins spectaculaire qu'avec une compression avec pertes, mais non négligeable (40% environ : cf. section 6.3).

De nombreux travaux tentent tout de même d'investir le cadre de la compression avec pertes pour l'imagerie médicale [Gau06, MC04, LLF03, WT01]. Ils essaient de quantifier la perte acceptable pour ne pas perturber le diagnostique. Des études récentes, comme celle de Koff *et al.* [KBB+08], s'attachent à définir des taux de compression pour lesquels les distorsions sont suffisamment faibles pour être tolérées lors de tests subjectifs. A l'aide de ce type d'investigations, l'Association Canadienne des Radiologistes (CAR) a également publié une norme visant à orienter les radiologues sur les taux de compression maximaux pouvant être utilisés avec les standards JPEG et JPEG2000 pour différentes modalités d'images et régions anatomiques étudiées [CAR08]. De telles études se focalisent sur des algorithmes particuliers et sont alors uniquement utilisables comme références pour d'autres travaux.

La plupart de ces études sont réalisées à l'aide de protocoles d'évaluation strictes, difficiles à mettre en place et coûteux[1]. Ces expertises portent souvent sur la qualité du diagnostique après compression. Ainsi, pour évaluer correctement un algorithme, il faut pouvoir réunir des images (de patients malades et sains) pour toutes les techniques d'acquisition (et de reconstruction) concernées et pour toutes les pathologies connues. Ces images doivent ensuite être évaluées par un panel de médecins spécialistes, pour différents taux de compression (et donc niveaux de dégradation). Pour s'astreindre des problèmes de l'évaluation subjective, la piste d'une métrique objective d'évaluation de la qualité par apprentissage sur des résultats d'experts a été envisagée [DRPV06]. Une telle approche, visant à « imiter » un expert, n'a, à notre connaissance, pas été utilisée afin d'optimiser les pertes en compression. Ceci pourrait pourtant permettre d'éviter l'introduction d'artefacts, de limiter la suppression de signes pathologiques (informations utiles aux diagnostiques) et de moins perturber l'habitude des médecins. On trouve tout de même quelques références à des travaux utilisant des modèles psychovisuels humains [KJML05], plus classiques en compression d'images naturelles.

1.3 Problèmes pouvant être rencontrés

Beaucoup de publications ne prennent pas en compte la précision d'acquisition des technologies récentes et proposent des résultats sur des images 8 bits. Ces images sont parfois natives, mais peuvent également être une quantification d'images de plus haute précision. Cette quantification est souvent effectuée comme une suppression des bits de poids faible, qui sont également les plus durs à compresser puisqu'ils sont les plus bruités. Les résultats ne sont donc pas réellement comparables à ceux des articles prenant en compte la précision complète.

De plus, et c'est souvent la cause du problème précédent, la plus part des codeurs d'images naturelles mais aussi des bibliothèques de traitement et des formats de stockage (de moins en moins vrai) ne prennent pas en compte des dynamiques supérieures à 8 bits, ce qui pose également des soucis pour comparer les algorithmes. Pour les codecs vidéo c'est encore plus souvent le cas.

Enfin comme il pourra être constaté dans le chapitre 6, les taux de compression d'un même algorithme varient énormément en fonction du matériel d'acquisition, de la technique de reconstruction utilisée et des éventuels post-traitements effectués sur l'image (filtrage, rehaussement de contraste, ...).

1.4 Bases d'images utilisées

Afin d'évaluer différents algorithmes, une sélection d'images volumiques natives et prétraitées a été retenue à partir de plusieurs sources :

- une base d'images[2] principalement destinée à l'évaluation des performances du logiciel OSIRIX[3], comportant environ 20 giga octets de données,
- des dossiers médicaux de patients,

[1] Ce référer au chapitre 5 de [NACM07, NACM08] pour plus de détails sur ces méthodes d'évaluation
[2] http://pubimage.hcuge.ch:8080
[3] http://www.osirix-viewer.com

- une base destinée à évaluer différents algorithmes de traitements d'images médicales : MeDEISA[4] (*Medical Database for the Evaluation of Image and Signal Processing Algorithms*),
- une base d'images en provenance directe du matériel d'acquisition : la base NLM-VHP[5] (*The National Library of Medicine's Visible Human Project*), comprenant deux corps complets (un Homme, une Femme) disponibles sous forme d'IRMs, de scanners, et de photographies couleurs de cryosections (ces dernières n'ont pas été utilisées).
- et une base de séquences d'images sur 8 bits disponible sur le site du CIPR[6] (*Center for Image Processing Research*).

Les trois dernières bases sont utilisées pour comparer des algorithmes de compression dans quelques publications. Toutes les valeurs négatives ont été mises à zéro. Celles-ci apparaissent parfois sur certaines tomographies (cas de MeDEISA) pour signaler les zones de l'image ne contenant aucune information reconstruite (extérieur du disque de reconstruction).

La sélection a permis de retenir des volumes avec une résolution axiale fine : tomographies contenant un bruit inter-coupe corrélé ou non (filtrage et/ou techniques de reconstruction différentes), et des volumes avec une résolution plus faible, possédant également un bruit corrélé ou non : IRM et IRM3D natives.

Ainsi des volumes, allant de très corrélées à très peu, sont à disposition pour évaluer l'efficacité de diverses techniques de compression.

1.5 Plan

Les enjeux de la compression des TDMs et IRMs ont été présentés, ainsi que les contraintes liées au contexte médical. Dans la suite de ce document, un survol de l'existant sera effectué et quelques pistes de recherche seront abordées.

Pour introduire les notions élémentaires et théoriques utilisées en compression d'images, un chapitre sera tout d'abord consacré à la théorie de l'information et la compression de données. Le suivant présentera des notions complémentaires relatives à la compression de signaux. Ces concepts seront utilisés dans le quatrième chapitre qui effectuera un état de l'art, plutôt focalisé sans perte, des méthodes de compression d'images bidimensionnelles. Dans le cinquième chapitre, consacré à la compression d'images médicales, nous verrons que les techniques destinées aux images naturelles sont souvent utilisées telles-quelles sur les images médicales bidimensionnelles, et qu'elles sont également la source d'inspiration des approches volumiques. Enfin dans le dernier chapitre nous présenterons les résultats de quelques expérimentations.

[4] http://www.medeisa.net
[5] http://www.nlm.nih.gov/research/visible/visible_human.html
[6] http://www.cipr.rpi.edu/resource/sequences/sequence01.html

Chapitre 2

Théorie de l'information et compression

Introduction

Avant d'aller plus loin, il peut être nécessaire de rappeler que si l'on souhaite effectuer une compression de données à l'aide d'un algorithme particulier, il doit exister un dual de cet algorithme permettant de décompresser l'information : on parle souvent de codec (codeur/décodeur). Par la suite il sera fréquent de ne voir des références qu'aux méthodes de compression, la méthode de décompression en découlant directement.

Pour un bon nombre des codecs, le décodeur possède une complexité similaire à celle du codeur, son algorithme peut être vu comme l'inverse de celui de compression. On parle alors de codec « symétrique ». Il existe également des procédés pour lesquelles le codeur détermine des paramètres qui seront directement transmis et n'auront donc pas besoin d'être recalculés lors du décodage. Généralement ces paramètres servent à réduire la quantité totale de l'information à transmettre et ne sont pas estimables (de manière identique) par le décodeur lorsqu'ils dépendent de données qui ne seront connues qu'après la décompression (information non causale). Dans ce cas on parle de codec « asymétrique ».

Bien que les algorithmes de compression d'images suivent souvent des schémas plus complexes que ceux de compression de données, la théorie et les principes restent similaires. Ce chapitre à donc pour but de présenter la théorie sur laquelle s'appuie tout système de compression, ainsi que les principes algorithmiques les plus couramment utilisés. La première section sera donc dédiée à introduire diverses définitions ainsi que les bases de la théorie permettant de déterminer les limites de la compression. La suivante présentera les algorithmes les plus connus qui permettent de s'approcher voir d'atteindre cette limite, à condition de connaître les statistiques de l'information à compresser. Une dernière section illustrera leur application en compression de données.

2.1 Théorie de l'information

La théorie de l'information définit les fondements mathématiques de la compression moderne. Elle fut introduite en 1948 par Claude Shannon [Sha48] en démontrant la limite de la compression de données numériques. Cette limite, appelée entropie, s'exprime comme le nombre moyen de symboles équiprobables nécessaires pour représenter un message provenant d'une source aléatoire d'information dont on connait le modèle statistique.

Les fondements de cette théorie seront présentés dans cette section.

2.1.1 Définitions

Un ALPHABET $\mathcal{A}$ est un ensemble, dont la LONGUEUR, ou cardinalité, est $|\mathcal{A}| = \mathrm{Card}(\mathcal{A}) \neq 0$. Les éléments de cet alphabet $\{a_1, \ldots, a_{m=|\mathcal{A}|}\}$ sont appelés SYMBOLES et sont ordonnés.

Soit un alphabet $\mathcal{A}$, un MODÈLE probabiliste $\mathtt{M}$ est une fonction

$$\begin{array}{rccll} \mathtt{M}: & \mathcal{A} & \mapsto & [0,1] & \\ & a_i & \rightarrow & \mathrm{P}_{\mathtt{M}}(a_i), & \sum_i \mathrm{P}_{\mathtt{M}}(a_i) = 1, \end{array} \tag{2.1}$$

qui associe une probabilité $\mathrm{P}_{\mathtt{M}}(a_i)$ à chaque symbole a_i de $\mathcal{A}$. Cette probabilité peut être une estimation et n'est donc pas nécessairement la probabilité réelle $\mathrm{P}(a_i)$ du symbole.

Soit un symbole $a_i \in \mathcal{A}$ en provenance d'une variable aléatoire X supposée suivre une loi de probabilité correspondant à un modèle $\mathtt{M}$. D'après le théorème de Shannon, la quantité d'information nécessaire à exprimer a_i à l'aide d'un alphabet Ω de longueur W est

$$\mathcal{I}_{\mathtt{M}}(X = a_i) = -\log_W \mathrm{P}_{\mathtt{M}}(X = a_i). \tag{2.2}$$

Cette mesure correspond à la longueur de la séquence de symboles de Ω nécessaire à exprimer de manière optimale le symbole a_i, en concordance avec le modèle $\mathtt{M}$.

En utilisant le modèle de probabilité a priori $\mathtt{M}$, l'ENTROPIE $H_{\mathtt{M}}(X)$ de la variable aléatoire X se définit alors comme

$$H_{\mathtt{M}}(X) = \sum_{a_i \in \mathcal{A}} \mathrm{P}(X = a_i)\mathcal{I}_{\mathtt{M}}(X = a_i), \tag{2.3}$$

et correspond à la quantité moyenne d'information nécessaire pour coder un symbole de $\mathcal{A}$ dont les statistiques d'apparition sont décrites par le modèle $\mathtt{M}$.

Dans le cas d'un modèle parfait ($\mathrm{P}_{\mathtt{M}}(a_i) = \mathrm{P}(a_i), \forall a_i \in \mathcal{A}$), on obtient l'ENTROPIE DE SHANNON $H(X)$ qui correspond à la quantité moyenne d'information nécessaire pour coder un symbole de $\mathcal{A}$ de façon la plus optimale :

$$H(X) = -\sum_{a_i \in \mathcal{A}} \mathrm{P}(X = a_i) \log_W \mathrm{P}(X = a_i). \tag{2.4}$$

On définit l'ENTROPIE CONDITIONNELLE de X sachant Y par :

$$H(X|Y) = -\sum_{x,y \in \mathcal{A}} \mathrm{P}(X = x, Y = y) \log_W \mathrm{P}(X = x|Y = y). \tag{2.5}$$

L'ENTROPIE JOINTE de la variable aléatoire discrète formée par le couple (X,Y) se déduit de (2.4) :

$$H(X,Y) = -\sum_{x,y \in \mathcal{A}} \mathrm{P}(X = x, Y = y) \log_W \mathrm{P}(X = x, Y = y). \tag{2.6}$$

vérifiant

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y). \tag{2.7}$$

Ainsi

$$H(X,Y) \leq H(X) + H(Y), \quad H(X,Y) \geq H(X) \quad \text{et} \quad H(X,Y) \geq H(Y), \tag{2.8}$$

et $H(X,Y) = H(X) + H(Y)$ quand X et Y sont indépendantes.

En informatique on utilise le plus souvent $\Omega = \{0,1\}$ pour un codage binaire de l'information, et on exprime alors la quantité d'information en bits/symbole.

Un MESSAGE $\mathcal{M} = (X_t)_{t \in \{1,\dots,T\}}$ de taille T est la réalisation d'un ensemble de variables aléatoires ordonnées (processus) X_t à valeurs dans un alphabet $\mathcal{A}_t = \{a_{0,t}, \dots, a_{m_t,t}\}$, suivant une loi de probabilité $\mathcal{P}_{X_t}$ qui peut dépendre de la réalisation des autres variables.

Selon cette définition, une variable X_t d'un message $\mathcal{M}$ de longueur T aurait pour entropie

$$H(X_t) = -\sum_{(a_{i_1},\dots,a_{i_T}) \in \mathcal{A}_1 \times \dots \times \mathcal{A}_T} \mathrm{P}(X_1 = a_{i_1}, X_2 = a_{i_2}, \dots, X_T = a_{i_T}) \log_W \mathrm{P}\left(X_t = a_{i_t} | (X_k = a_{i_k})_{k \neq t}\right). \tag{2.9}$$

Clairement, cette définition ne convient pas pour la compression à cause de la dépendance qui existe entre les variables et qui rend le modèle non causal.

Classiquement, on modélise donc un message d'une façon beaucoup plus simple par un modèle de Markov d'ordre k. Ainsi la loi de probabilité de toutes variable aléatoire X_t ne dépend plus que des k variables qui la précèdent. On exprime alors l'entropie conditionnelle

$$H_k(X_t) = -\sum_{(a_{i_0},\dots,a_{i_k}) \in \mathcal{A}_t \times \dots \times \mathcal{A}_{t-k}} \mathrm{P}(X_t = a_{i_0}, \dots, X_{t-k} = a_{i_k}) \log_W \mathrm{P}\left(X_t = a_{i_0} | (X_{t-j} = a_{i_{t-j}})_{j \in \{1..k\}}\right). \tag{2.10}$$

En compression, les messages considérés sont finis et, pour que le traitement puisse être rendu causal, les modélisations des lois de probabilités $\mathcal{P}_{X_t}$ ne dépendent que d'informations connues : précédemment encodées (celles-ci contiennent éventuellement le(s) modèle(s) à utiliser), et/ou fixées au codeur et au décodeur.

2.1.2 Mesures

Dans la littérature, diverses mesures sont utilisables pour comparer des algorithmes de compression. On voit parfois apparaitre le DÉBIT ENTROPIQUE DE LA SOURCE (aussi appelé ENTROPIE DE LA SOURCE). Il peut se définir à l'aide de l'entropie jointe (2.6) pour une source $\mathcal{X}$,

$$H(\mathcal{X}) = \lim_{N\to\infty} \frac{1}{N} H(X_0, \ldots, X_{N-1}). \tag{2.11}$$

En compression d'image, on considère des messages $\mathcal{M}$ finis de longueur N. L'entropie est alors

$$H(\mathcal{M}) = \frac{1}{N} H(X_0, \ldots, X_{N-1}). \tag{2.12}$$

La source est l'image à compresser et les lois de probabilités sont calculées à partir du nombre d'occurrence de chaque symbole.

Il arrive souvent que l'entropie de la source exprimée soit en fait $H(X)$ (ou entropie d'ordre 0), en considérant que l'image est une succession de réalisations indépendantes d'une variable X dont la loi de probabilité est donnée par :

$$\mathrm{P}(a_i) = \frac{n_{a_i}}{N}, \tag{2.13}$$

où n_{a_i} est le nombre d'occurrences du symbole a_i dans le message.

2.2 Codage entropique

Un codeur entropique permet de compresser une séquence de symboles en se basant sur leur probabilité (a priori) d'apparition. Ainsi, ces symboles se voient assigner une nouvelle représentation (variable en fonction de leur probabilité d'apparition) de manière à ce que le codage de la source s'approche au plus de son entropie. Ces codeurs entropiques doivent être vus comme des codeurs permettant de compresser l'information en générant un code sans ambiguïté, et permettant donc de faire une restitution sans perte. Il existe des techniques dérivées de celles présentées ici permettant de dépasser la limite entropique en induisant des ambiguïtés et donc un codage avec pertes, mais elles ne seront pas abordées.

Les deux sous sections suivantes seront consacrées à deux approches différentes pour la construction de codes à longueur variable (VLC : *Variable Length Code*). Les VLC sont très utilisés car ils sont simples à mettre en place, rapides et souvent efficaces. Un codeur VLC fonctionne de la façon suivante :

Pour tout symbole $a_{\mathcal{A}_\mathcal{I},i}$ de l'alphabet d'entrée $\mathcal{A}_\mathcal{I}$ de longueur $N_\mathcal{I}$, le codeur VLC $\mathcal{K}_\mathtt{M}$, conditionné par un modèle $\mathtt{M}$, associera une unique séquence de $N_{\mathtt{M},i}$ symboles de l'alphabet de sortie $\mathcal{A}_\mathcal{O}$, appelée mot. On définit ainsi un alphabet $\mathcal{AS}_\mathcal{O}$ pour lequel chaque symbole est une séquence de symboles de l'alphabet $\mathcal{A}_\mathcal{O}$:

$$\mathcal{AS}_\mathcal{O} = \{a_{\mathcal{AS}_\mathcal{O},i}\}_{i\in\{1..N_\mathcal{I}\}} / a_{\mathcal{AS}_\mathcal{O},i} \in {\mathcal{A}_\mathcal{O}}^{N_{\mathtt{M},i}}. \tag{2.14}$$

Ainsi, un tel codeur $\mathcal{K}_\mathtt{M}$ peut être représenté comme une fonction qui, à un symbole dans un contexte donné (modèle $\mathtt{M}$), associera un unique mot (codage « $1 \to 1$ »), cette association est bijective afin de permettre le décodage :

$$\begin{array}{rccc} \mathcal{K}_\mathtt{M} : & \mathcal{A}_\mathcal{I} & \mapsto & \mathcal{AS}_\mathcal{O} \\ & a_{\mathcal{A}_\mathcal{I},i} & \to & a_{\mathcal{AS}_\mathcal{O},i}. \end{array} \tag{2.15}$$

L'avantage majeur de tels codeurs est leur rapidité. Leur principal inconvénient est qu'ils sont souvent sous optimaux puisqu'ils ne proposent que des séquences de symboles $a_{\mathcal{AS}_\mathcal{O},i}$ de longueurs entières.

La troisième sous section abordera donc un autre type de codeur entropique très prisé pour ses performances : le codeur arithmétique. Il permet de compresser une séquence de symboles ${\mathcal{A}_\mathcal{I}}^N$ en une séquence de symboles ${\mathcal{A}_\mathcal{O}}^M$ de telle sorte que chaque symbole d'entrée ne soit pas nécessairement associé à un nombre entier de symboles de l'alphabet de sortie. Un tel codeur permet toujours de rester très proche de l'entropie si le modèle statistique $\mathtt{M}$ est correct.

2.2.1 Code de Huffman

Le code de Huffman, mis au point en 1952 [Huf52], est l'un des premiers à avoir émergé et est donc l'un des plus répandus. Son principe est d'attribuer un code court à un symbole fréquent et un code plus long à un symbole plus rare. En se plaçant dans le cas d'un alphabet de sortie binaire, l'attribution du nombre de bits pour chaque symbole se fait par le biais d'un arbre binaire, construit en fonction de la fréquence d'apparition des symboles.

A chaque nœud de l'arbre est associé la fréquence d'apparition de l'ensemble des symboles présents dans son sous arbre. L'arbre est construit de manière ascendante, à partir de ses feuilles. Chaque feuille correspond à un symbole, et à sa fréquence d'apparition. Itérativement, jusqu'à l'obtention de la racine, un nœud père est généré pour le couple des deux nœuds ayant la fréquence d'apparition la plus faible.

Le code de chacun des symboles est alors le chemin partant de la racine jusqu'à la feuille lui correspondant (bit 0 pour aller au fils gauche, bit 1 pour aller au fils droit). Ce code est défini de telle sorte qu'aucun des symboles de l'alphabet $\mathcal{AS}_\mathcal{O}$ ne soit le préfixe d'un autre : chaque symbole étant associé à une feuille de l'arbre le chemin qui permet d'y accéder ne peut en aucun cas être le sous-chemin d'un autre. Ainsi, le codage est non ambigu.

La création de cet arbre peut être effectuée à l'aide d'un algorithme de complexité $\mathrm{O}(N_\mathcal{I} \log_2 N_\mathcal{I})$, avec $N_\mathcal{I}$ la taille de l'alphabet.

Le code Huffman peut être statique, transmis ou adaptatif. Lorsqu'il est statique, le code est fixé pour le codeur et le décodeur et n'a donc pas besoin d'être transmis. Cependant, il n'est pas toujours adapté aux données.

Lorsqu'il est transmis, l'arbre doit être construit par le codeur à partir des fréquences d'apparition des symboles : un parcours supplémentaire des données est donc nécessaire (sur la totalité, ou un échantillon représentatif). Il a également un coût pour la compression puisqu'il doit être transmis.

Enfin lorsqu'il est adaptatif, en partant d'un arbre particulier les fréquences des symboles sont mises à jour au fur et à mesure de leur apparition. L'arbre doit donc être mis à jour régulièrement, ce qui demande un temps de calcul important. L'arbre de départ peut également être statique ou transmis.

Avantages :

- la rapidité (le plus rapide, une fois l'arbre binaire généré) : avec une table indexée par les symboles d'entrée, le temps de codage dépend uniquement de la recopie (dans le message de sortie) de la séquence associée au symbole d'entrée.
- ce codage est optimal pour les modèles où $\forall a_{\mathcal{A}_\mathcal{I},i} \in \mathcal{A}_\mathcal{I}, -\log_W \mathrm{P}_\mathtt{M}(a_{\mathcal{A}_\mathcal{I},i})$ est une valeur entière (i.e. $\forall a_{\mathcal{A}_\mathcal{I},i} \in \mathcal{A}_\mathcal{I}, \exists k \in \mathbb{N} / \mathrm{P}_\mathtt{M}(a_{\mathcal{A}_\mathcal{I},i}) = W^{-k}$), avec W la longueur de l'alphabet de sortie $\mathcal{A}_\mathcal{O}$.

Inconvénients :

- un modèle où $\mathrm{P}_\mathtt{M}(a_{\mathcal{A}_\mathcal{I},i}) = W^{-k}$ est peu courant. Dans toutes autre situation, le codage de Huffman s'éloigne de l'entropie en associant à chaque symbole $a_{\mathcal{A}_\mathcal{I},i}$ une séquence de longueur $\lceil -\log_W \mathrm{P}_\mathtt{M}(a_{\mathcal{A}_\mathcal{I},i}) \rceil$.
  Il reste tout de même efficace lorsque $\mathrm{P}_\mathtt{M}(a_{\mathcal{A}_\mathcal{I},i}) \simeq W^{-k}$
- le modèle `M` est difficilement adaptatif. Ceci nécessite un recalcul complet de l'alphabet de sortie $\mathcal{AS}_\mathcal{O}$ à chaque phase d'adaptation, ce qui est coûteux en temps de calcul. Pour palier à ce problème, on peut utiliser le contexte pour sélectionner un modèle prédéfini et le codage Huffman associé, par exemple.

2.2.2 Code de Golomb

Dans le principe, un code Golomb est assez similaire à un code Huffman. La principale différence vient du fait qu'on suppose que les variables d'entrée suivent une certaine loi de probabilité qui va implicitement induire le codage. Cette propriété permet de ne pas avoir besoin de calculer la table de correspondances des symboles, et le rend ainsi facilement adaptatif [WSS96][Mal06].

Un codeur Golomb [Gol66] de paramètre m noté G_m encode une valeur entière positive n en deux parties : une représentation binaire de $(n \mod m)$ et une représentation unaire de $\lfloor n/m \rfloor$. Les codeurs Golomb sont optimaux pour coder des valeurs entières positives, suivant une loi de probabilité géométrique de la forme $Q(n) = (1-\rho)\rho^n$ avec $0 < \rho < 1$ pour une valeur $m = \lceil \log(1+\rho)/\log(\rho^{-1}) \rceil$.

Les symboles de l'alphabet $\mathcal{AS}_\mathcal{O}$ sont construits par la concaténation de la représentation binaire de taille fixe de $(n \mod m)$ et de la représentation unaire de taille variable de $\lfloor n/m \rfloor$. Pour que le codage soit non ambigu, la représentation unaire est une suite de bits à 1 suivie d'un bit d'arrêt à 0.

Les codeurs Golomb-Rice sont un cas particulier du codeur Golomb avec $m = 2^k$, particulièrement adaptés pour le codage/décodage informatique. Ainsi les divisions se font par simple décalage de bits et les modulo par une opération logique utilisant un masque.

Avantages :
- la rapidité (légèrement moins que Huffman) : il nécessite uniquement un modulo pour obtenir les symboles représentant la partie de poids faible des symboles d'entrée, et d'une division suivie du codage unaire (une table comme pour Huffman pourrait être utilisée, mais la propriété suivante serait perdue),
- il est facilement rendu adaptatif : seul le paramètre m (ou k) nécessite d'être estimé,
- ce codage est optimal pour des modèles ou les valeurs suivent une loi géométrique, ce qui est une bonne approximation de la répartition des valeurs à compresser dans les schémas destinés aux signaux ou aux images.

Inconvénients :
- Tout comme Huffman, il y a une perte entropique du à l'encodage VLC (« $1 \rightarrow 1$ »). Cette perte peut cependant être diminuée à l'aide de systèmes adaptatifs [WSS96].

Ce codeur et ses dérivés sont très présents dans la littérature pour le codage d'erreurs de prédiction en compression sans perte d'images et de signaux audio, domaines pour lesquels il a tendance à être plus utilisé qu'Huffman puisqu'il nécessite beaucoup moins de calculs pour être rendus adaptatif. En effet l'adaptation du code Huffman nécessite le recalcul de l'arbre complet ($\mathrm{O}(N_{\mathcal{I}} \log_2 N_{\mathcal{I}})$), tandis que pour le code Golomb seul le paramètre m doit être mis à jour ($\mathrm{O}(1)$). Les valeurs à encoder ont également tendance à suivre une loi géométrique en valeur absolue, ce qui permet au code Golomb d'être efficace.

2.2.3 Codage arithmétique

Le codeur arithmétique tente de compresser l'information et d'atteindre le débit entropique de la source. Il permet de coder un symbole de l'alphabet d'entrée en un nombre fractionnaire de symboles dans l'alphabet de sortie et évite ainsi le problème des codeurs VLC. Il est alors quasiment optimal au sens de l'entropie du modèle statistique `M` utilisé.

Un tel codeur permet de générer un message de sortie $\mathcal{M}_{\mathcal{O}}$ en un temps proportionnel à la taille du message d'entrée $\mathcal{M}_{\mathcal{I}}$ (complexité linéaire). Cependant il nécessite des traitements plus complexes que ceux d'un codeur VLC qui le rendent plus lent.

Le principe consiste à coder toute une séquence de symboles en un unique nombre décimale dont la précision permet de regénérer la séquence originale. Ce nombre correspond à la borne inférieure d'un intervalle dans lequel la séquence est plongée. On part d'un intervalle $I_0 = [x_0 = 0, y_0 = 1)$. Soit une séquence de symboles $(X_t)_{t \in 1..T}$, appartenant à l'alphabet $\mathcal{A}_{\mathcal{I}} = \{a_1 \dots a_{N_{\mathcal{A}_{\mathcal{I}}}}\}$, dont la distribution statistique est modélisée par le modèle `M`. Si I_k est l'intervalle obtenu après le codage du k-ième symbole, I_k est subdivisé en $N_{\mathcal{A}_{\mathcal{I}}}$ sous-intervalles de longueurs $\mathtt{M}(a_i)\|I_k\|$, proportionnelles aux probabilités d'apparition de chaque symbole. L'intervalle I_{k+1} correspondra alors au X_{k+1}-ième sous-intervalle associé au symbole à coder.

Ainsi l'intervalle I_0 est subdivisé et réduit symbole après symbole, en fonction du modèle, de manière non ambiguë en un intervalle $I_N = [x_N, y_N)$. x_N est progressivement codé (affiné) après chaque subdivision k avec une précision permettant de distinguer x_k de y_k, et rend donc également le codage non ambiguë.

Avantages :
- ce codage permet d'atteindre des taux très proches de l'entropie théorique,
- il est facilement adaptatif, sans avoir à modifier l'algorithme de compression lui même, mais le modèle qu'il utilise.

Inconvénients :
- le codage arithmétique demande un peu plus de temps de calcul que les codeurs VLC : ces derniers sont quasiment instantanés puisqu'ils ne nécessitent que d'une lecture en mémoire du code et d'une recopie, tandis que le codeur arithmétique doit effectuer quelques opérations (au minimum 2 multiplications, 1 division, 3 additions et 2 soustractions pour chaque symbole dans l'implémentation de [BCK07]) afin de mettre à jour les bornes de l'intervalle,

– le codage est très dépendant des symboles précédemment encodés, ce qui le rend peu robuste aux erreurs de transmission, et empêche le décodage à partir d'une position aléatoire dans le flux de données.

On peut trouver de bons supports pour comprendre et programmer un tel codeur tels que celui d'Amir Said [Sai04] ou celui d'Eric Bodden, Malte Clasen et Joachim Kneis [BCK07] (code source inclus).

Il existe de nombreux codeurs arithmétiques, tous basés sur le même principe, mais visant à offrir des réponses à ses inconvénients. Certains cherchent à réduire le temps de calcul en proposant une légère perte de l'optimalité du codage. D'autres essaient d'inclure des informations redondantes afin de permettre une correction d'erreur plus aisée en cas de pertes d'informations durant la transmission.

Comme ils permettent la représentation des symboles sur un nombre fractionnaire de bits, ils sont bien adaptés à la compression de messages ayant un petit alphabet. Un bon nombre de codeurs arithmétiques binaires ont donc été développés, parmi lesquels on peut citer le Q-Coder d'IBM [PMLA88] qui utilise son propre modèle adaptatif et qui réduit le temps de calcul en supprimant les multiplications par des approximations, au détriment d'une légère perte de compression. Il est l'ancêtre du QM-Coder utilisé dans JBIG et JPEG (JPEG peut aussi utiliser un autre codeur dérivé appelé Q15 [ITU05]) et du MQ-Coder utilisé par JBIG2 et JPEG2000 [MYRP07]. On peut également citer celui d'Amir Said FastAC[1] (*Fast Arithmetic Coding*) ou encore CABAC (*Context-Adaptive Binary Arithmetic Coding* utilisé par H.264/MPEG-4 AVC.

2.2.4 Autres approches

Les codes VLC et arithmétiques sont issus d'un modèle statistique explicite : les probabilités "a priori" des occurrences des symboles sont fournies au codeur. Une autre classe de codeurs entropiques se distingue également, pour laquelle le modèle statistique est implicite : les probabilités sont indirectement exploitées. Deux exemples répandus seront présentés ici.

L'un des codages les plus simples, et très utilisé, est le RLE (*Run Length Encoding*). Il est très efficace dans les situations où l'on sait que de longues séquences d'un même symbole doivent survenir. Il consiste simplement à spécifier le symbole et la longueur de la séquence. Cette technique est utilisée en compression d'image en compléments des codes VLC afin de palier à leur manque d'efficacité à compresser ces longues séquences de symboles identique. Le plus souvent, un symbole spécial est utilisé pour préciser le changement de mode (VLC vers RLE), mais celui-ci peut également se faire de manière automatique en fonction du contexte local comme dans LOCO-I (cf. 4.1.3).

Le codage par substitution est une autre approche utilisée en compression de données. Il est souvent très efficace sur du texte et certains types d'images. Cette technique consiste à générer un dictionnaire de portions du message qui sont redondantes, et remplacer leurs occurrences par leur index dans le dictionnaire. Cette technique est apparue avec l'algorithme LZ77 [ZL77] qui est désormais utilisé en compression sans perte d'images par le format PNG. Ce format fut créé afin de contrecarrer sa variante propriétaire LZW (Lempel Ziv Welch) utilisée dans les fichiers GIF (*PNG's Not Gif*).

2.3 Compression de données brutes

Les algorithmes de compression de données brutes sont génériques et cherchent à obtenir de bons taux de compression quel que soit le type de source fournie. Dans cette section, sera tout d'abord abordée la notion de taux de compression, permettant d'évaluer les performances d'un algorithme de compression. Ensuite, quelques approches classiques et schémas de compression de données usuels seront présentés.

2.3.1 Taux de compression

Soit l'alphabet $\mathcal{A}_\mathcal{I}$ de longueur $N_\mathcal{I}$ du message d'entrée $\mathcal{M}_\mathcal{I}$ et $\mathcal{A}_\mathcal{O}$ de longueur $N_\mathcal{O}$ celui du message de sortie $\mathcal{M}_\mathcal{O}$. Si la longueur de $\mathcal{M}_\mathcal{I}$ est $L_\mathcal{I}$ et celle de $\mathcal{M}_\mathcal{O}$ est $L_\mathcal{O}$, on peut arbitrairement les exprimer en nombre de bits afin de les rendre comparables : $L_{b\mathcal{I}} = L_\mathcal{I} \log_2 N_\mathcal{I}$ est le nombre de bits nécessaires à décrire $\mathcal{M}_\mathcal{I}$ et $L_{b\mathcal{O}} = L_\mathcal{O} \log_2 N_\mathcal{O}$ le nombre de bits nécessaires à décrire $\mathcal{M}_\mathcal{O}$.

[1] http://www.cipr.rpi.edu/~said/FastAC.html

On exprime souvent le taux de compression T à l'aide d'une notation du type "$T : 1$". Il correspond au ratio du gain d'espace réalisé par le processus de codage :

$$T = \frac{L_{b\mathcal{I}}}{L_{b\mathcal{O}}} \tag{2.16}$$

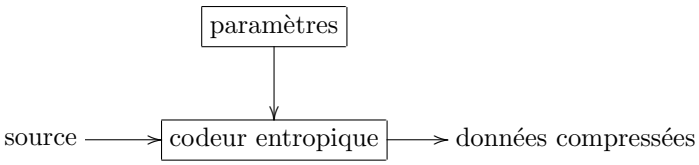
$T : 1$ peut être lu comme « T pour 1 » et interprétée comme : « T bits d'entrée peuvent être représentés par un unique bit en sortie »

En compression d'image il est plus courant d'exprimer la compression en terme de débit, soit en bits par pixel (bpp). Si $\mathcal{M}_{\mathcal{I}}$ est une image possédant $P_{\mathcal{I}}$ pixels, cette mesure est alors :

$$\text{bpp} = \frac{L_{b\mathcal{O}}}{P_{\mathcal{I}}} \tag{2.17}$$

2.3.2 Approches usuelles

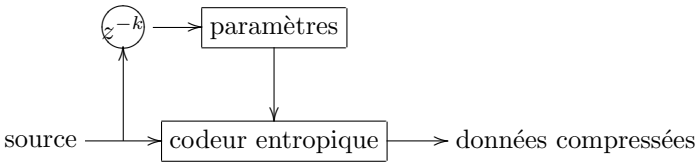
Puisque la compression de données est prévue pour fonctionner quel que soit le type d'information fournie, de tels codeurs ne font pas d'hypothèses sur l'organisation de l'information. Les algorithmes ont donc globalement des schémas relativement simples pouvant se résumer parfois à un simple codage entropique :



(2.18)

Un codeur VLC ou arithmétique à paramètres statiques ou encore l'algorithme LZ77, par exemple, peuvent être appliqués en suivant ce schéma.

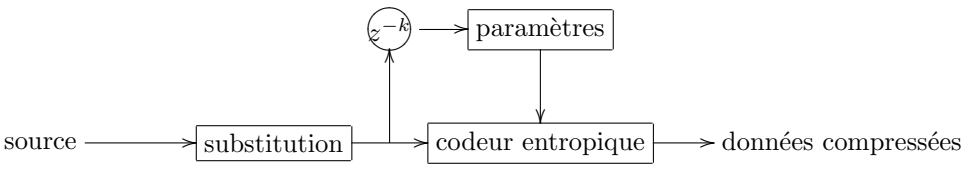
Des codeur adaptatifs simples peuvent également être produit :



(2.19)

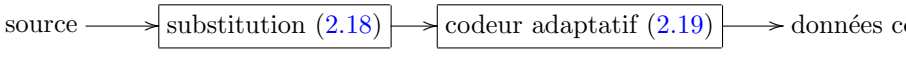
Le plus souvent, $k = 1$. Ce modèle classique est très fréquemment utilisé en tant que composant interne d'un schéma de compression de signaux.

L'algorithme LZMA (Lempel Ziv Markov chain Algorithm) est l'un des algorithmes de compression les plus performants. Il utilise le principe de dictionnaire du LZ77 qui, bien qu'efficace, produit un message de substitution dans lequel subsiste toujours une certaine redondance. Ce message de substitution est donc compressé à l'aide d'un codeur entropique. LZMA utilise un codeur arithmétique et une modélisation statistique effectuée à l'aide de chaînes de Markov. Le schéma utilisé correspond alors à :



(2.20)

qui peut être représenté comme l'enchaînement d'un codeur simple (2.18) suivi d'un codeur adaptatif (2.19) :



(2.21)

Comme on peut le constater, les systèmes de compression de données sont modélisables à l'aide de schémas très simples. Cette simplicité est inhérente à l'objectif de tels algorithmes, à savoir rester efficace quel que soit le type d'information à compresser, et se généralise pour les systèmes les plus performants et les plus utilisés.

Bien que considérés comme profitables dans un cadre d'utilisation général, ils le sont souvent beaucoup moins dans des cadres plus spécifiques. En effet, bien que les schémas soient élémentaires, les algorithmes

peuvent nécessiter d'une quantité de mémoire et de calculs considérables, ce qui peut les rendre inexploitables par des systèmes embarqués ou temps réels. Une utilisation des composants plus respectueuse pour leur environnement d'exécution est alors requise, au détriment de pertes de performances. Enfin lorsqu'un algorithme est destiné à compresser un type de données particulier, des méthodes mieux adaptées et donc plus efficaces sont envisageables. C'est notamment le cas pour la compression de signaux ou d'images.

Conclusion

Dans ce chapitre nous avons pu présenter la limite entropique de la compression pouvant être déterminée à l'aide des statistiques de la source, ainsi que les algorithmes permettant de l'atteindre ou de s'en approcher. Cette limite est souvent estimée avec l'entropie d'ordre 0, qui considère toutes les valeurs du message comme des réalisations indépendantes de variables suivant une même loi. Elle peut être dépassée avec l'aide de modèles markoviens et/ou adaptatifs estimant les statistiques locales du message ou avec l'aide de schémas appropriés au type de données.

Les schémas de compression de données brutes que nous avons abordés restent très génériques, ne faisant aucune hypothèse sur les données manipulées, et ne sont donc pas optimaux. Des algorithmes dédiés à des données respectant certaines propriétés peuvent donc être développés et se montrer plus efficaces. C'est notamment le cas pour la compression d'images, et plus généralement pour la compression de signaux pour lesquels une certaines marge d'erreur et également parfois tolérée. Le chapitre suivant sera donc destiné à présenter les principes et outils utilisés en compression de signaux.

Chapitre 3

Compression de signaux

Introduction

Les schémas simples de compression de données, bien qu'efficaces pour des données hétérogènes, ne sont pas adaptés pour la compression de signaux. En effet, les données des signaux numériques sont presque toujours corrélées (l'entropie d'ordre 0 du signal n'est pas représentative de l'entropie de la source) et peuvent être représentées avec des précisions plus ou moins importantes pouvant aller de 1 à 32 bits (parfois plus), les valeurs pouvant être signées ou non, avoir une représentation entière ou en virgule flottante, et posséder plusieurs canaux entrelacés. Cependant les systèmes de compression de données utilisent souvent un alphabet de taille fixe (256 valeurs pouvant être représentées sur un octet par exemple) et considèrent que le flux de données est unique. Ce modèle trop simple peut provoquer de grosses pertes de performances.

Certaines applications permettent également d'approximer les signaux sous la contrainte de certains critères visant à minimiser les distorsions. La compression de ces approximations revient à effectuer une compression avec pertes (*lossy compression*) et permet d'atteindre des taux très intéressants.

Ce chapitre sera dédié à l'introduction des concepts et des outils utilisés en compression des signaux et plus particulièrement des images. Dans une première section nous introduirons les pertes liées à la numérisation de signaux pour enchaîner sur les notions de compression avec pertes, ainsi que les techniques usuelles permettant d'effectuer ces pertes et d'optimiser leur influence. Nous terminerons en abordant quelques notions complémentaires qui sont intéressantes pour la représentation des signaux et leur manipulation une fois compressés.

3.1 Signaux numériques

Beaucoup de signaux sont issus de phénomènes physiques, et plus généralement d'ondes (compression/décompression de matériaux pour les ondes sonores, ondes électro-magnétiques pour les photographies, ...). Ces ondes peuvent être modélisées à l'aide de fonctions (ou variables aléatoires) continues à valeurs réelles. Prenons pour exemple une fonction $f(t)$ telle que $f : \mathbb{R}^+ \mapsto \mathbb{R}$, et considérons t comme le temps. Sur un intervalle de temps $[t_0; t_1]$ ($t_0 \neq t_1$), aussi petit qu'il soit, f possède une infinité de réalisations. De même pour un instant t donnée $f(t)$ a une précision numérique infinie, puisqu'à valeur réelle.

Les supports numériques ayant une capacité limitée (nombre de bits), ces fonctions à précision infinie ne peuvent pas être stockées directement. Ainsi la numérisation n'est pas immédiate, il faut discrétiser et limiter le nombre d'instants auxquels on souhaite conserver f mais également le nombre de valeurs prises par $f(t)$.

La discrétisation (ou échantillonnage) de $[t_0, t_1]$ se fait le plus souvent à pas fixe T. On conservera alors les valeurs $\{F(t_0),\ F(t_0+T),\ F(t_0+2T),\ \ldots\}$, où $F : \mathbb{R}^+ \mapsto \mathbb{R}$ est issue de f (par exemple $F(t) = f(t)$, ou encore $F(t) = \frac{1}{T}\int_t^{t+T} f(u)\,du$). Cette échantillonnage est effectué habituellement de manière à respecter le théorème de Nyquist-Shannon : le signal F ne doit posséder aucune composante fréquentielle au delà de la plage $[-\frac{1}{2}S_F; \frac{1}{2}S_F]$ où $S_F = \frac{1}{T}$ est appelé fréquence d'échantillonnage. Cette contrainte est suffisante pour pouvoir restituer les valeur $F(t)$ quel que soit t.

La discrétisation des valeurs prises par $F(t)$ s'effectue ensuite à l'aide d'une fonction $Q : \mathbb{R} \mapsto \mathcal{A}_\mathcal{Q}$, où $\mathcal{A}_\mathcal{Q}$ est l'alphabet des valeurs autorisées pour leur représentation numérique. Cette opération n'est autre qu'une quantification effectuée sur un alphabet de taille infinie (cf. section suivante).

Suite à ces discrétisations, la représentation numérique d'un signal réel est donc imparfaite et cause des pertes d'informations. Un compromis doit donc être fait entre la précision du signal et le nombre de bits nécessaires pour sa représentation.

3.2 Pertes et quantification

Afin d'augmenter les performances de compression des signaux numériques, on peut chercher à diminuer le nombre de symboles de l'alphabet (i.e. les valeurs numériques autorisées par la représentation discrète) en entrée du processus de codage entropique. Cette diminution peut s'effectuer en associant à tout symbole de l'alphabet de départ un nouveau symbole dans un alphabet plus petit, ce qu'on appelle la quantification scalaire.

On peut également construire un dictionnaire réduit de séquences de symboles représentatives (qui nécessitera d'être transmis si il ne peut être construit de façon similaire au codeur et au décodeur) et associer à chaque séquence d'entrée du système l'indexe de la séquence la plus similaire dans le dictionnaire. On appelle cette approche la quantification vectorielle. Le codeur entropique n'est alors utilisé que pour compresser les indexes.

Dans ces deux situations, on effectue une opération surjective. Les symboles ou séquences ainsi transformés ne pourront jamais être récupérés, il existe donc des pertes d'information qui sont optimisables.

3.2.1 Quantification scalaire

Soit $\mathcal{A}_\mathcal{I}$ l'alphabet initialement utilisé pour représenter le signal et $\mathcal{A}_\mathcal{Q}$ l'alphabet après quantification, tels que $\mathrm{Card}(\mathcal{A}_\mathcal{Q}) < \mathrm{Card}(\mathcal{A}_\mathcal{I})$, l'opérateur de quantification peut s'écrire :

$$\begin{array}{rccc} Q: & \mathcal{A}_\mathcal{I} & \mapsto & \mathcal{A}_\mathcal{Q} \\ & x & \to & x_q \end{array}, \tag{3.1}$$

et l'operation de dequantification pseudo inverse :

$$\begin{array}{rccc} \bar{Q}: & \mathcal{A}_\mathcal{Q} & \mapsto & \mathcal{A}_\mathcal{I} \\ & x_q & \to & \tilde{x} \end{array}. \tag{3.2}$$

Par la suite, la valeur approximée $\tilde{x} = \bar{Q}\left(Q(x)\right)$ sera notée $\tilde{Q}(x)$.

Exemple — 3.2.1 (*Un quantificateur scalaire simple*)

Si l'alphabet initial $\mathcal{A}_\mathcal{I}$ est l'ensemble des valeurs entières pouvant être représentée sur N bits, $\mathrm{Card}(\mathcal{A}_\mathcal{I}) = 2^N$. Un exemple simple de quantification scalaire est la réduction de la précision des échantillons du signal. On peut ainsi décider de supprimer un certain nombre de bits (k) afin de réduire la taille de l'alphabet. L'alphabet des valeurs quantifiées $\mathcal{A}_\mathcal{Q}$ sera alors l'ensemble des valeurs entières pouvant être représentée sur $N-k$ bits soit $\mathrm{Card}(\mathcal{A}_\mathcal{Q}) = 2^{N-k}$. La manière la plus simple d'effectuer cette réduction est un décalage de bits supprimant ceux de poids faible. L'opération pseudo-inverse consiste alors à remplacer les bits supprimés par des bits nuls par exemple.

Puisque l'application Q est surjective, on peut chercher à minimiser l'erreur de quantification $|x-\tilde{Q}(x)|$ selon certains critères. A titre d'exemple, l'algorithme Lloyd-Max [Max60, Llo82] cherche la solution optimale au problème de minimisation de l'espérance de l'erreur quadratique due à la quantification scalaire :

$$\mathrm{E}\left[\left(X - \tilde{Q}(X)\right)^2\right].$$

Cependant, utilisée dans un codec, ce type d'optimisation nécessite que le décodeur connaisse également la loi de probabilité de X. Les paramètres de sa distribution (ou un dictionnaire de symboles) doivent donc être transmis (ou fixés).

Comme il sera présenté par la suite (cf. section 3.3), les signaux sont de préférence décorrélés avant d'être quantifiés. Leur représentation après décorrélation est souvent propice à l'utilisation d'un simple quantificateur uniforme. Ce dernier consiste à proposer des valeurs quantifiées $\left\{\tilde{Q}(\mathcal{A}_\mathcal{Q})\right\}$ uniformément réparties dans $\mathcal{A}_\mathcal{I}$.

Un autre quantificateur très répandu est le quantificateur uniforme avec zone morte qui possède une zone de quantification plus importante autour de l'origine et reste uniforme partout ailleurs.

Notons également que le principe de quantification est utilisé pour la représentation finie des valeurs réelles en virgule flottante. Cette discrétisation de l'ensemble des valeurs est souvent la cause d'accumulations d'erreurs de calculs, et de pertes d'informations lorsque des transformations numériques à valeurs réelles sont employées. La transformée de Fourier discrète ou celle en cosinus qui sera présentée dans la section 3.3 sont de bons exemples.

3.2.2 Quantification vectorielle

Loin d'être optimale pour des problèmes tels que la compression, la quantification scalaire peut être améliorée en considérant un alphabet initiale $\mathcal{A}_{\mathcal{V}_\mathcal{I}}$ dont chaque symbole est une combinaison (ou vecteur) de symboles de l'alphabet $\mathcal{A}_\mathcal{I}$. La quantification vectorielle (QV) permet de réduire les distorsions (en terme d'erreur quadratique) introduites par une simple quantification scalaire uniforme.

Les systèmes de compression utilisant ce type de quantification sont souvent asymétriques : le codeur doit estimer les paramètres optimaux de quantification et les transmettre. Un dictionnaire optimal est ainsi construit. Chacun de ses éléments est un représentant de l'alphabet $\mathcal{A}_{\mathcal{V}_\mathcal{I}}$ qui sera utilisé par la suite pour reconstruire le signal. Les valeurs quantifiées correspondent alors à l'index du représentant du dictionnaire le plus similaire selon une métrique de distorsion donnée (cf. section suivante).

Il existe de nombreuses méthodes pour construire le dictionnaire. L'un des algorithmes les plus utilisés est celui de Linde, Buzo et Gray (LBG) [LBG80, GG91], qui est similaire à la méthode des k-means pour le clustering de données. LBG est une généralisation en dimension finie de l'algorithme Lloyd-Max. Il part d'un dictionnaire sous-optimal et cherche à l'améliorer. Sa complexité pour trouver un dictionnaire optimal est exponentielle avec la dimension des vecteurs, et sa convergence dépend du dictionnaire initial. Il existe donc des techniques sous optimales, visant à offrir de bonnes performances pour un temps de calcul plus faible, la technique la plus rapide étant de fixer une partition uniforme par exemple.

3.2.3 Théorie débit-distorsion

Comme mentionné précédemment la numérisation des signaux pose le problème de trouver le débit binaire nécessaire à obtenir une distorsion donnée et inversement. De même, lorsqu'une compression avec pertes est effectuée, les principaux objectifs peuvent être de compresser au maximum tout en respectant une certaine qualité, ou encore d'introduire le minimum de distorsions pour un taux de compression fixé.

La théorie débit-distorsion cherche à apporter des solutions aux deux problèmes précédents. Pour un signal f donné, soit $\mathtt{R}(Q)$ le débit pour une configuration des paramètres de compression Q (généralement le quantificateur) et $\mathtt{D}(Q)$ la distorsion, alors la fonction débit-distorsion $D(R_m)$ est définie de manière théorique comme la solution au problème de minimisation de la distorsion pour un débit maximal R_m, et la fonction distorsion-débit $R(D_m)$ est définie comme celle du plus petit débit pour une distorsion maximale D_m. Ces problèmes d'optimisation sous contrainte peuvent s'écrire :

$$D(R_m) = \min_{Q\in\mathcal{Q},\, \mathtt{R}(Q)\leq R_m} \mathtt{D}(Q),$$

$$R(D_m) = \min_{Q\in\mathcal{Q},\, \mathtt{D}(Q)\leq D_m} \mathtt{R}(Q).$$

avec $\mathcal{Q}$ l'ensemble des configurations possibles des paramètres de compression.

Si R et D sont des fonctions convexes, ces problèmes sont souvent résolus par une minimisation lagrangienne sans contrainte :

$$J_D(Q,\lambda) = \mathtt{D}(Q) + \lambda\mathtt{R}(Q),$$

$$J_R(Q,\lambda) = \mathtt{R}(Q) + \lambda\mathtt{D}(Q),$$

avec $\lambda > 0$.

Ainsi, une métrique de qualité et/ou de distorsion $\mathtt{D}(Q)$ adaptée au contexte applicatif ou aux contraintes de calculs est utilisée afin d'optimiser le débit versus les distorsions.

Mesures de distorsion

De nombreuses métriques peuvent être utilisées pour mesurer les distorsions. La majorité de celles-ci se fondent sur la norme de Minkowski :

$$\mathrm{Err} = \left(\sum_k |e_k|^\beta \right)^{1/\beta},$$

où e_k est l'erreur entre le k-ième élément du signal f et sa version estimée $\tilde{f}$:

$$e_k = f(k) - \tilde{f}(k).$$

La plus utilisée pour l'optimisation débit-distorsion est l'erreur quadratique moyenne (MSE : *Mean Squared Error*). Pour un signal de longueur N, elle s'exprime comme

$$\mathrm{MSE} = \frac{1}{N} \sum_{k=0}^{N-1} \left| f(k) - \tilde{f}(k) \right|^2.$$

Les valeurs de la MSE étant souvent très élevées à bas débit, le PSNR (*Peak-to-peak Signal-to-Noise Ratio*) qui s'exprime en décibel (dB) est souvent préférée pour comparer des courbes de distorsions :

$$\mathrm{PSNR}_{\mathrm{dB}} = 10 \log_{10} \frac{(2^p - 1)^2}{\mathrm{MSE}},$$

où p est le nombre de bits de précision des échantillons. Dans le cas de la compression sans perte, le PSNR est infini, et il décrois selon l'erreur.

La minimisation de l'erreur quadratique moyenne revient à minimiser les variations d'énergies entre le signal original et celui reconstruit. Cependant, cette variation énergétique n'est pas toujours adaptée, principalement lorsque le résultat de la compression est destiné à être apprécié par des humains (images, sons, ...) qui n'ont pas une sensibilité globale (moyenne des erreurs sur la totalité de l'image, du signal), mais plus localisée (variations spatiales, fréquentielles, temporelles...). Ainsi des métriques visant à simuler le comportement du cerveau humain face aux distorsions sont utilisées pour optimiser la compression : des modèles psycho-acoustiques pour la compression sonore, tels que celui utilisé pour la norme internationale MP3 (MPEG-1/2 Audio Layer 3 : ISO/CEI 13818-3), et des modèles psycho-visuels (HVS *Human Vision System*) pour l'image et la vidéo, dont les plus connues sont la métrique JND (*Just Noticeable Difference*) de Lubin (Sarnoff Corporation) [Lub93, Lub95, Lub97, LF97], et la VDP (*Visible Difference Predictor*) de Daly [Dal93, Dal94].

Ces modèles psycho-visuels utilisent souvent les trois principales variations auxquelles l'œil humain est sensible : la sensibilité à l'intensité lumineuse, à la fréquence spatiale (contrastes), et au contenu (effets de masquage fréquentiel). Le masquage, également utilisé dans les modèles psycho-acoustiques, est un effet produit en présence de plusieurs fréquences d'intensité différentes : l'humain perçoit moins les plus faibles (dans des zones texturées, ou proches de contours contrastés par exemple).

Il existe également d'autres modèles moins basés sur des études de la perception mais tout de même efficaces (en particulier sur des images naturelles) telles que celles proposées par Wang : SSIM (*Structural SIMilarity*) [WBSS04] et MS-SSIM (*MultiScale SSIM*) [WSB03] une extension multi-échelle plus robuste.

Ces métriques demandent souvent une complexité calculatoire assez importante, qui les rend difficilement utilisable avec des systèmes de compression rapides. Wang utilise tout de même SSIM dans [WLS07] pour effectuer l'optimisation d'un codage progressif et obtient des résultats intéressants.

Compression presque sans perte

Dans certaines situations, on va plutôt vouloir autoriser une variation sur chaque échantillon qui soit inférieure à un seuil δ donné. Dans ce cas, si $s(k)$ est le signal original et $\tilde{s}(k)$ le signal détérioré par la compression (ou signal approché), alors on cherche à obtenir :

$$\forall k \quad |s(k) - \tilde{s}(k)| \leq \delta \qquad \Longleftrightarrow \qquad \|s - \tilde{s}\|_\infty \leq \delta \quad .$$

Cette approche est souvent qualifiée de compression presque sans perte (*near lossless*).

Pour obtenir le seuil δ minimal satisfaisant un taux de compression donné, le PAE (*Peak of Absolute Error*)

$$\mathrm{PAE} = \max_k |s(k) - \tilde{s}(k)| = \|s - \tilde{s}\|_\infty$$

est utilisé comme métrique pour l'optimisation débit-distorsion.

Comme il sera présenté dans la section suivante, les signaux peuvent être décorrélés à l'aide de méthodes prédictives, ou de transformées. Pour les méthodes prédictives, la prédiction est effectuée à l'aide des informations déjà quantifiées (et vérifiant un PAE donné). L'erreur de prédiction peut alors aisément être quantifiée à son tour pour respecter le même PAE. Un quantificateur scalaire uniforme avec un pas de quantification égal au δ souhaité sera alors le plus pratique.

Dans des schémas par transformée, le PAE est beaucoup plus contraignant que la MSE. En effet si la transformée est orthonormale, la MSE liée à une quantification sera la même dans l'espace transformé que dans l'espace original ; et si la transformée est biorthogonale, elle reste simple à estimer. Cependant, si l'on prend pour exemple une décomposition en sous-bandes et que l'on cherche a respecter un PAE $= \delta$, une quantification scalaire uniforme de pas δ sur une seule des sous-bande impliquera déjà un PAE égal à δ sur le signal reconstruit et interdira alors la quantification uniforme des autres bandes. Des méthodes plus appropriées doivent donc être mises en place (cf. section 6.4).

3.3 Décorrélation des signaux

Comme brièvement mentionné dans la section précédente, les échantillons composant les signaux sont en général corrélés et possèdent donc une information redondante. Une étape de décorrélation visant à supprimer cette redondance s'avère alors très bénéfique pour l'amélioration des taux de compression. Ce traitement s'effectue en amont de la quantification pour être plus efficace. Les deux grandes familles visant à effectuer cette décorrélation seront présentées dans cette section, à savoir les approches par prédiction et celles par transformation.

3.3.1 Prédiction

La décorrélation par prédiction, souvent généralisé sous le nom de codage DPCM (*Differential Pulse Code Modulation*), consiste à estimer la valeur probable $\hat{x}_i$ d'un échantillon x_i à l'aide d'une information causale et à ne conserver que l'erreur de cette prédiction :

$$\epsilon_i = x_i - \hat{x}_i$$

L'information causale est généralement composée de l'ensemble des échantillons précédemment décorrélés $x_{j\in\{0..i-1\}}$ (ainsi que de toutes valeur pouvant être calculées à partir de celles-ci), mais peut être différente à condition que le processus de prédiction puisse être inversé pour reconstruire le signal original (données précédemment transmises).

3.3.2 Transformation

La transformation consiste à changer l'espace de représentation des données. Pour la compression, il est préférable qu'elle ait de bonnes propriétés de décorrélation et nécessaire qu'elle soit inversible de manière à pouvoir restituer le signal par la suite. La fonction $T : s \rightarrow \widehat{s}_\mathsf{T}$ permettant de changer l'espace de représentation de tout signal s est appelée transformée. Lorsqu'elle est inversible, il existe une fonction $T^{-1}(\widehat{s}_\mathsf{T})$, appelée transformée inverse, telle que $T^{-1}(T(s)) = s$. Ne seront abordée ici que les transformées sur des signaux discrets finis (signaux échantillonnés et de longueur finie). Pour ces signaux, deux classes de transformées seront distinguées : les transformées redondantes et non redondantes. Si T est redondante, alors $\mathrm{Card}(T(s)) > \mathrm{Card}(s)$. En compression sans perte on préfère souvent le cas non redondant : $\mathrm{Card}(T(s)) = \mathrm{Card}(s)$.

Certaines transformations sont théoriquement inversibles, mais ne le sont pas totalement dans la pratique : des erreurs d'arrondis peuvent survenir lors de calculs sur des valeurs flottantes. D'autres peuvent également nécessiter un nombre infini de coefficients dans l'espace transformé pour être inversibles. Dans la pratique, on peut utiliser ces transformées et limiter le nombre de coefficients pour obtenir une approximation du signal. Dans les deux cas, la compression sera considérée comme une compression avec pertes.

Dans cette section, la transformée en cosinus discrets (DCT *Discret Cosine Transform*) et la transformée en ondelettes discrète (DWT *Discrete Wavelet Transform*), qui sont les plus utilisées en compression, seront tout d'abord présentées de manière pratique. Ces deux transformées sont détaillées de façon théorique dans le livre de référence de Stéphane Mallat [Mal08]. D'autres représentations, moins présentes dans la littérature, mais également utilisées en compression, seront ensuite mentionnée.

DCT

Issue des travaux de Joseph Fourier, la transformée en cosinus continue *CT* permet de représenter une fonction continue f, définie sur un intervalle borné $[a, b]$), à l'aide d'une combinaison linéaire de cosinusoïdes. Elle est très connue en traitement du signal depuis que sa version discrète *DCT* a été proposée [ANR74]. Plusieurs bases de cosinus peuvent être construites à partir des séries de Fourier (appelées bases de cosinus I, II, III, IV, ...). Elles diffèrent par la représentation du signal $\tilde{f}$ généré par extension en dehors de l'intervalle $[a, b]$.

Pour un signal s de longueur N les DCT-II et DCT-IV se définissent comme :

$$\widehat{s}_{\mathsf{DCT-II}}(k) = \sum_{n=0}^{N-1} s(n) \cos\left(\frac{\pi\left(n+\frac{1}{2}\right)k}{N}\right),$$
$$\widehat{s}_{\mathsf{DCT-IV}}(k) = \sum_{n=0}^{N-1} s(n) \cos\left(\frac{\pi\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)}{N}\right),$$

pour $k \in [0..N-1]$. Et leurs transformées inverses sont identiques à un facteur de normalisation près :

$$s(n) = \frac{2}{N}\left[\sum_{k=0}^{N-1} \widehat{s}_{\mathsf{DCT-II}}(k) \cos\left(\frac{\pi\left(k+\frac{1}{2}\right)n}{N}\right)\right],$$
$$s(n) = \frac{2}{N}\left[\sum_{k=0}^{N-1} \widehat{s}_{\mathsf{DCT-IV}}(k) \cos\left(\frac{\pi\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)}{N}\right)\right],$$

La DCT-II est utilisée sur des blocs distincts en compression d'images (après une application bidimensionnelle de la transformée). Elle suppose que le signal généré par extension est $2N$-périodique et est pair en $-\frac{1}{2}$ et en $N-\frac{1}{2}$ et donc $s(k) = s(-k-1)$ pour $k < 0$ et $s(k) = s(2N-1-k)$ pour $k \geq N$.

En compression audio (codecs MP3 ou AC-3 par exemple) c'est une version modifiée de la DCT-IV, la MDCT [PJB87] , qui est souvent utilisée. Elle est conçue pour être appliquée sur des portions du signal ayant un recouvrement. Ainsi, pour un signal s, les portions p_i de longueur $2N$ seront telles que p_i aura ses N premiers échantillons en commun avec p_{i-1}, et ses échantillons suivant en commun avec p_{i+1}.

$$\widehat{p}_{i\mathsf{MDCT}}(k) = \sum_{n=0}^{2N-1} p_i(n) \cos\left(\frac{\pi\left(n+\frac{1}{2}+\frac{N}{2}\right)\left(k+\frac{1}{2}\right)}{N}\right)$$

pour $k \in [0..N-1]$. Et sa transformée pseudo-inverse :

$$\tilde{p}_i(n) = \frac{1}{N}\sum_{k=0}^{N-1} \widehat{p}_{i\mathsf{MDCT}}(k) \cos\left(\frac{\pi\left(n+\frac{1}{2}+\frac{N}{2}\right)\left(k+\frac{1}{2}\right)}{N}\right)$$

pour $n \in [0..2N-1]$. Elle a la propriété de fournir un nombre de coefficients transformés qui est deux fois moins important que le nombre d'échantillons. Ce nombre de coefficients réduits implique une erreur de reconstruction. Cependant cette erreur est supprimée par l'ajout de l'information obtenue par recouvrement des portions. Ce phénomène est connu sous le nom de TDAC (*time-domain aliasing cancellation*).

Au final, pour qu'un signal s de longueur $M = gN$ puisse être reconstruit parfaitement, p_0 devra contenir N échantillons arbitraires (nuls ou symétrie), puis les N premiers échantillons de s, et le dernier bloc p_g contiendra les N derniers échantillons de s et N autres arbitraires. Ainsi la transformation complète du signal fournira $(g+1)N$ coefficients, soit une légère redondance.

Contrairement à la DCT-II, la DCT-IV suppose une représentation impaire en $N-\frac{1}{2}$. Ainsi lorsque le bloc transformé n'est pas nul en ce point, une discontinuité est créée, et des coefficients de forte amplitude apparaissent. C'est pourquoi en compression on préfère appliquer une fenêtre avant d'utiliser la MDCT. Dans ce cas, pour que la MDCT puisse être inversible quelques opérations supplémentaires seront nécessaires.

Ondelettes

La DCT et plus généralement les transformées de Fourier permettent de représenter un signal fini comme une somme de fonctions sinusoïdales (et donc de longueurs infinies) en supposant que celui-ci est périodique en dehors de ses bornes de représentation. Contrairement à ces approches, les ondelettes permettent une représentation localisée, temps/fréquence, du signal en le projetant sur des fonctions oscillantes à support borné, appelées ondelettes. Ces ondelettes sont issues d'une ondelette mère Ψ (ici normalisée $\|\Psi\| = 1$ et centrée au voisinage de $t = 0$) par décalage temporel (translatée de $u \in \mathbb{Z}$ pour travailler sur des signaux discrets) et par dilatation (facteur d'échelle $s \in \mathbb{N}$ pour obtenir une représentation discrète), et forment une famille :

$$\mathcal{W} = \left\{ \Psi_{u,s}(t) = \frac{1}{\sqrt{2^s}} \Psi \left(\frac{t-u}{2^s} \right) \right\},$$

le facteur $\frac{1}{\sqrt{2^s}}$ permettant de normaliser $\|\Psi_{u,s}\| = 1$. Pour $t \in \mathbb{R}$, Ψ doit vérifier :

$$\int_{-\infty}^{+\infty} \Psi(t)\, dt = 0,$$

et les ondelettes discrétisées formant la famille $\mathcal{W}_{\mathbb{Z}}$:

$$\sum_{t \in \mathbb{Z}} \Psi_{u,s}(t) = 0.$$

La transformée en ondelettes à un instant u et à une échelle s d'un signal discret f est alors la projection de f sur $\Psi_{u,s}$:

$$\widehat{f}_{\mathsf{DWT}}(u,s) = \sum_{t \in \mathbb{Z}} f(t) \Psi_{u,s}(t),$$

et peut être vu comme un produit de convolution :

$$\widehat{f}_{\mathsf{DWT}}(u,s) = f \star \bar{\Psi}_s(u),$$

avec

$$\bar{\Psi}_s(t) = \frac{1}{\sqrt{2^s}} \Psi \left(\frac{-t}{2^s} \right).$$

La transformée en ondelettes permet ainsi de modéliser un filtrage multi-échelle.

Pour un signal discret f fini de longueur N, $\widehat{f}_{\mathsf{DWT}}(u,s)$ sera défini pour $u \in [0..N-1]$. La DWT peut alors avoir quelques variantes en fonction de comment est modélisée l'extension du signal lorsque le filtrage nécessite des valeurs de f au delà de l'intervalle $[0..N-1]$. On peut par exemple considérer une symétrisation, une périodicité, ou des valeurs nulles.

Afin de pouvoir reconstituer le signal original à partir d'une décomposition effectuée jusqu'à une échelle $s = j$, il est nécessaire de rajouter les basses fréquences correspondantes aux échelles supérieures à j. On peut alors introduire la fonction d'échelle Φ (aussi appelée ondelette père) qui permet de représenter le signal f à différentes échelles en agissant comme un filtre passe bas. Elle est définie par :

$$\Phi(t) = \int_{1}^{+\infty} \Psi_{0,s}(t)\, ds,$$

qui permet également de générer une famille de fonctions d'échelles :

$$\mathcal{S} = \left\{ \Phi_{u,s}(t) = \frac{1}{\sqrt{2^s}} \Phi \left(\frac{t-u}{2^s} \right) \right\}.$$

La représentation basses fréquences du signal f peut alors s'écrire :

$$\widehat{f}_{\Phi}(u,s) = f \star \bar{\Phi}_s(u),$$

avec

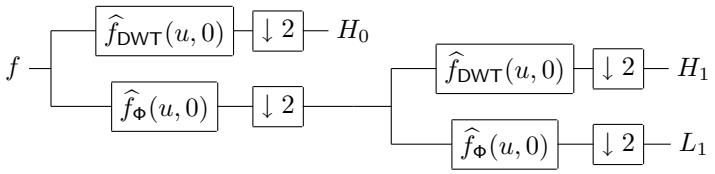
$$\bar{\Phi}_s(t) = \frac{1}{\sqrt{2^s}} \Phi \left(\frac{-t}{2^s} \right).$$

Pour être appliquée en compression sans perte, la familles d'ondelettes discrètes $\mathcal{W}_{\mathbb{Z},u\in[0..N-1],s\leq j}$ doit être choisie de manière à former une frame. Ainsi il existera une frame duale $\mathcal{W}^*_{\mathbb{Z},u\in[0..N-1],s\leq j} = \left\{\Psi^*_{u,s}\right\}$ telle que :

$$f(t) = \sum_{u=0}^{N-1} \widehat{f}_{\Phi}(u,j)\Phi^*_j(t) + \sum_{s=0}^{j}\sum_{u=0}^{N-1} \widehat{f}_{\mathsf{DWT}}(u,s)\Psi^*_{u,s}(t)$$

On parlera alors d'ondelettes bi-orthogonales, et les Φ et Ψ seront appelés filtres d'analyse, tandis que les Φ^* et Ψ^* seront appelés filtres de synthèse. Et si $\mathcal{W}$ est orthogonale, alors $\mathcal{W}^* = \mathcal{W}$.

Jusqu'à présent la décomposition est redondante : les $\widehat{f}_{\Phi}(u,0)$ ainsi que chacun des j niveaux de décomposition $\widehat{f}_{\mathsf{DWT}}(u,j)$ nécessitent chacun N coefficients. Cependant, les $\widehat{f}_{\mathsf{DWT}}(u,0)$ correspondant aux hautes fréquences du signal selon Ψ, et les $\widehat{f}_{\Phi}(u,0)$ aux basses fréquences, un sous-échantillonnage dans chacune des bandes peut être effectué. On peut ensuite ré-appliquer la décomposition sur les basses fréquences sous-échantillonnées. La décomposition revient alors à effectuer une analyse en sous-bandes par cascade de bancs de filtres :

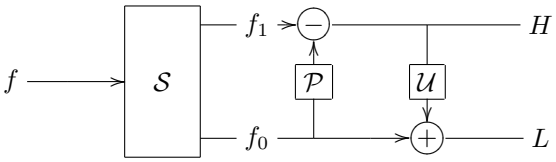


(3.3)

où $\downarrow 2$ correspond à un sous-échantillonnage uniforme d'un facteur 2.

Il n'est pas toujours évident de calculer analytiquement les filtres de synthèse permettant de reconstruire le signal d'origine. Cependant, une technique connue sous le nom de *lifting scheme*, dont les concepts fondamentaux et des références à d'autres articles sont disponibles dans [Swe96], permet de concevoir très facilement des filtres d'analyse et leur filtre dual.

Le principe repose sur un partitionnement $\mathcal{S}$ (comme *split*) des échantillons du signal f en deux ensembles f_0 et f_1 (ou plus ; simplifié à deux pour illustrer). f_0 est alors utilisé pour effectuer la prédiction $\mathcal{P}$ (comme *predict*) des échantillons de f_1, qui se voient ré-attribuer leur erreur de prédiction et peuvent alors être considérés comme les hautes fréquences H sous-échantillonnées de f. Cette erreur de prédiction est ensuite utilisée pour effectuer une mise à jour $\mathcal{U}$ (comme *update*) sur f_0 consistant à supprimer ces hautes fréquences identifiées dans H et à produire un signal basses fréquences sous-échantillonné L :



(3.4)

Le schéma peut ensuite être réitéré sur les basses fréquences pour obtenir une décomposition dyadique, et sur les hautes fréquences pour obtenir une décomposition en paquets d'ondelettes.

Ce schéma de lifting est couramment employé par les algorithmes de compression pour sa simplicité algorithmique et donc sa vitesse d'exécution et pour son faible coût mémoire, puisque la transformée peut ainsi être effectuée dans le même espace mémoire que le signal initial.

Les ondelettes ont une propriété intéressante pour la compression : si Ψ a ses P premiers moments nuls (régularité d'ordre p), alors, la projection de tout polynôme d'ordre inférieur à P sur Ψ sera nulle :

$$\int_{-\infty}^{+\infty} x^p \Psi(x)\, dx = 0 \quad \forall p \in \mathbb{Z}/P$$

Ainsi plus Ψ sera régulière, plus la transformée génèrera des coefficients nuls (et réduira ainsi l'entropie).

Malheureusement, le nombre de moments nuls est limité par le support de l'ondelette. Plus la régularité sera élevée et plus le nombre d'oscillations, nécessaires à l'annulation des moments, le sera également. Daubechies a montré que pour obtenir une ondelette orthogonale à P moments nuls, le filtre associé à l'ondelette nécessite d'au moins $2P$ coefficients. Les ondelettes de Daubechies [Dau88, Dau92] ont été créées pour posséder ces P moments nuls et avoir un support le plus compact possible.

Or, plus la taille du support est élevée, plus l'ondelette va capter des discontinuités éloignées(polynômes d'ordre important). Dans ce cas, la projection aura des valeurs non nulles.

Un compromis entre le nombre de moment nuls et la taille du support doit donc être fait. En compression sans perte d'images, on utilise souvent les « ondelettes spline 5/3 » de Le Gall [LGT88] (ce qui

signifie que le filtre passe bas possède 5 coefficients et que le passe haut en possède 3) qui correspondent à un lifting par spline (partitionnement pair/impair, prédiction des échantillons impairs comme appartenant aux droites passant par leurs deux voisins pairs, mise à jour par ajout aux échantillons pairs de la moitié de la moyenne des erreurs commises sur leurs deux voisins impairs). Tandis qu'en compression avec pertes on préfère souvent les « ondelettes CDF 9/7 » de Cohen, Daubechies et Feauveau [CDF90].

Comme mentionné dans l'introduction de la section, le livre de référence de Stéphane Mallat [Mal08] permettra d'approfondir la théorie et les applications de cette transformée.

Autres

Bien que la DCT et la DWT soient les plus utilisées en compression, il en existe beaucoup d'autres, dont des techniques dérivées : la DPWT (*Discret Packet Wavelet Transform*), par exemple, consiste à réappliquer des décompositions sur les bandes hautes fréquences de façon à mieux décorréler les coefficients d'ondelettes.

La KLT (*Karhunen-Loève Transform*), équivalente à la PCA (*Principal Component Analysis*), est réputée pour son optimalité théorique en terme de débit-distorsions pour la compression de sources suivant une loi gaussienne. Pour les sources suivant une loi quelconque, cette transformée orthonormale permet d'obtenir la meilleur décorrélation possible, et minimise l'erreur quadratique moyenne (MSE) qu'il est possible d'obtenir avec un nombre de coefficients restreints.Cependant, pour que ces optimalités soient vérifiées, il faut connaître les vecteurs propres engendrés par la décomposition (et donc les transmettre) et la complexité calculatoire est assez importante : il faut calculer la matrice de covariance, effectuer sa décomposition en valeurs propres et vecteurs propres, et effectuer la projection des données sur la base ainsi obtenue. La DCT est, dans le cas de processus de Markov gaussiens, réputée pour en être une bonne approximation des vecteurs propres de sa KLT, elle est donc préférée pour sa plus faible complexité. Le lecteur pourra se référer au chapitre 4 de [TM01] ainsi qu'aux articles [PTMO07, BSS09].

La WHT (*Walch-Hadamard Transform*), également connue sous le nom de transformée en "S" ou "ondelette de Haar", a eu une utilisation accrue en compression sans perte, et se voit encore utilisée aujourd'hui pour son support compact et donc la faible dépendance entre ses coefficients [WQ05, DBM06]. Elle fut améliorée en compression sans perte d'images par une étape de prédiction permettant de mieux décorréler ses coefficients : transformée "S+P" [SP93, SP96a].

Une autre transformation très connue et qui a pu être utilisée en compression sans perte et presque sans perte d'images est la pyramide laplacienne [BA83, AABL97, ABA01]. Celle ci est construite récursivement par la création d'une image basses fréquences sous-échantillonnée, et une image hautes fréquences pleine résolution, correspondant à l'erreur entre l'image basses fréquences interpolée pour fournir une image pleine résolution, et l'image originale. Ainsi, elle est redondante d'un facteur 4/3, et n'est donc pas beaucoup utilisée en compression.

On peut également citer les polynômes orthogonaux de Legendre (qui ont également donné lieux aux ondelettes de Legendre) qui peuvent être utilisés en compressions de signaux et d'images ; la transformée de Radon discrète, connue sous le nom de transformée "Mojette" pour laquelle il y a eu quelques essais en compression sans perte d'images [KA08]. Cette transformée non redondante ne prend qu'un nombre très limité d'orientations (3 par exemple) pour effectuer la projection de l'image.

Il existe encore une panoplie de transformées, certaines efficaces pour la compression, d'autres moins, d'autres encore inconnues du domaine... Leur efficacité dépend principalement de leurs propriétés, de celles du signal et de celles souhaitées dans le schéma de compression (avec ou sans perte, représentation progressive (cf. section suivante), ...).

3.4 Propriétés complémentaires des schémas de compression

Bien que le taux de compression soit un critère d'évaluation important, les cadres applicatifs peuvent requérir une certaine flexibilité de la part des codecs. Ces propriétés jouent un rôle important sur le choix des techniques à mettre en place et sur l'organisation du flux de données compressées. Les principales fonctionnalités que cherchent à offrir les formats de fichiers sont : l'accès aléatoire, la progressivité et la représentation sous forme d'objets. Ces propriétés sont utiles pour faciliter la manipulation du contenu et visent à réduire les couts de transfert, de calculs et de stockage mémoire.

3.4.1 Accès aléatoire

L'accès aléatoire aux données offre la possibilité d'accéder rapidement à une information ou une plage d'informations dont on connaît la localisation dans l'espace de représentation du signal une fois décompressé (positionnement temporel, spatial, spatio-temporel, ...). La compression de l'information

rend l'accès aléatoire immédiat impossible. Cependant, l'ajout d'une information permettant la localisation de différentes zones est un procédé usuel et permet un accès aléatoire grossier qui peut être suivi d'une décompression jusqu'à obtention des données souhaitées.

Ainsi, les formats de fichiers qui cherchent à offrir un accès aléatoire sont conçus à l'aide d'un en-tête et/ou de marqueurs dans le flux qui apportent des repères de localisation. Lorsqu'un codage adaptatif est utilisé, les modèles statistiques doivent être réinitialisé à chaque repère de localisation afin de permettre un décodage de chaque segment indépendamment des autres. Cette réinitialisation ainsi que l'information de localisation entrainent des pertes de compression.

La mise en place de l'accès aléatoire a donc un coût, et plus la précision d'accès sera fine, plus il faudra s'attendre a voir le débit augmenter. Cependant il permet d'éviter le transfert et la décompression de la totalité de l'information précédent la portion souhaitée.

3.4.2 Progressivité

Contrairement a une organisation séquentielle qui permet de reconstruire le signal échantillon par échantillon (ou bloc d'échantillons par blocs d'échantillons), la progressivité apporte une représentation de plus en plus précise du signal tout au long du flux de données. On parle souvent de représentation *scalable* ou encore *lossy to lossless* lorsque cette progressivité permet d'atteindre une représentation sans perte à la fin de la décompression. On distingue deux principaux types de progressivité : celle en qualité et celle en résolution.

Qualité progressive

La qualité progressive est synonyme de réduction successive d'erreurs d'approximations du signal. Celui-ci est d'abord transmis avec une faible qualité/précision (beaucoup de pertes/distorsions), puis est progressivement raffiné (réduction des distorsions) à l'aide de nouvelles informations.

Equitz et Cover [EC91] démontrent qu'une telle représentation sous forme d'arbre de raffinages successifs ne peut être optimale qu'à la condition que les différentes étapes de raffinage puissent s'écrire sous la forme d'une chaîne de Markov. Ainsi, la compression de signaux sous une forme progressive ne doit pas espérer atteindre des taux supérieurs aux approches non progressives. Cependant, de tels algorithmes ont l'avantage de permettre d'arrêter le codage ou le décodage dès qu'une contrainte de débit ou de distorsion (voir de temps de calcul) est atteinte, sans forcément augmenter la complexité de l'algorithme. Ainsi cette propriété permet d'obtenir un résumé de l'information rapidement.

Résolution progressive

Lorsqu'une résolution progressive est utilisée, le flux offre une représentation multi-échelle du signal qui est tout d'abord codé à basse résolution et suivi par l'information nécessaire pour produire une représentation plus fine à une échelle supérieure.

La résolution progressive peut être couplée à une qualité progressive, de manière à réduire progressivement les distorsions pour chacune des résolutions. Elle peut également être considérée, seule, comme une représentation en qualité progressive puisqu'il est possible d'interpoler une représentation à basse afin de produire un signal de plus haute résolution, les distorsions locales étant les erreurs d'interpolation.

3.4.3 Objets et régions d'intérêt

Dans certaines situations, on souhaite distinguer de façon indépendante différentes zones d'information (pas forcément régulières), afin d'y accéder aléatoirement, d'ordonner leur décompression, de favoriser leur progressivité, ou encore pour leur attribuer des qualités différentes pour une compression avec pertes. On considère ainsi ces zones comme des objets qui ont leurs propres propriétés de compression (si plusieurs algorithmes ou bases de transformation ont été mis en concurrence par exemple) et/ou peuvent posséder des informations supplémentaires de natures diverses (méta-données).

Lorsqu'une telle approche est envisagée, il faut prévoir une partie du débit pour la définition de la localisation des objets d'intérêt dans le signal, pour leur localisation dans le flux de données afin de permettre un accès aléatoire, et éventuellement pour contenir leurs méta-données.

Conclusion

Ce chapitre a effectué un tour d'horizon des techniques employées en compression des signaux et permettant d'obtenir des taux meilleurs qu'avec les algorithmes génériques du chapitre précédent. Nous avons commencé par présenter ce que sont les signaux numériques et la façon dont ils sont numérisés. Ceci a permis en particulier d'introduire les notions de compression avec pertes d'informations, et les bases de la théorie débit-distorsion qui si rapporte. Nous avons ensuite abordé les deux grandes familles d'algorithmes utilisés pour décorréler l'information présente dans ces signaux : la prédiction et la transformation. Enfin nous avons pu voir quelques propriétés pouvant permettre une utilisation plus efficace des fichiers compressés, à savoir l'accès aléatoire, la progressivité, et le découpage en objets. Ces propriétés peuvent cependant se révéler contraignantes et entraîner des baisses de performances sur la compression.

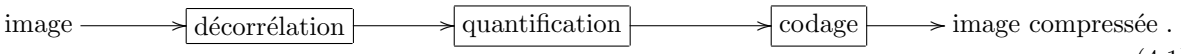
Les outils présentés ici seront réutilisés et spécialisés en compression d'images dans le chapitre suivant.

Chapitre 4

Compression d'images

Introduction

Comme présenté dans le chapitre précédent (cf. chapitre 3), le schéma général d'un processus de compression de signaux peut se résumer en une étape de décorrélation suivie d'un procédé de quantification (si on souhaite introduire des pertes) et d'un codage entropique pour finir. Ce schéma est également valable pour les images qui sont des signaux bidimensionnels particuliers :



(4.1)

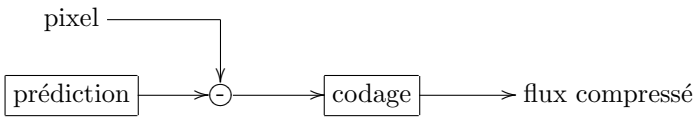
Dans ce chapitre sera dressé un état de l'art, plutôt focalisé sans perte, des techniques de compression qui peuvent satisfaire les besoins en imagerie médicale. Il sera organisé en deux sections dédiées au codage prédictif et par transformée.

4.1 Codage prédictif

Cette section présentera les trois algorithmes prédictifs les plus connus en compression d'images : celui utilisé pour la compression sans perte dans le standard JPEG, son successeur LOCO-I qui fut choisi pour normaliser JPEG-LS, et enfin CALIC qui est souvent employé comme algorithme de référence. L'évolution des approches plus récentes sera abordée dans une dernière partie.

4.1.1 Schéma général

Bien que JPEG-LS intègre également la possibilité d'introduire quelques erreurs afin d'améliorer la compression (cette extension sera brièvement abordée), tous ces algorithmes sont destinés à effectuer une compression sans perte. Ces codeurs sont appliqués en une seule passe sur l'image, ligne par ligne et colonne par colonne. Leur schéma de fonctionnement se décrit a un niveau du pixellique. On peut ainsi considérer que les pixels sont compressés un à un :



(4.2)

4.1.2 JPEG sans perte

Le standard JPEG (ISO/IEC International Standard 10918-1 – ITU-T Recommendation T.81) défini un mode sans perte utilisant un processus de prédiction (non basé sur la DCT, contrairement au mode avec pertes).

Les pixels sont parcourus séquentiellement, ligne après ligne. Pour rester causal, chaque pixel est prédit à l'aide de ses voisins précédemment encodés. Si x_i est le symbole à prédire, JPEG utilise le motif suivant :

c	b
a	x_i

Le standard propose 7 schémas pour effectuer la prédiction. Le schéma retenu étant utilisé sur la totalité de l'image et spécifié dans l'en-tête du fichier.

n°	prédiction
0	pas de prédiction
1	$\hat{x}_i \triangleq a$
2	$\hat{x}_i \triangleq b$
3	$\hat{x}_i \triangleq c$
4	$\hat{x}_i \triangleq a + b - c$
5	$\hat{x}_i \triangleq a + ((b - c)/2)$
6	$\hat{x}_i \triangleq b + ((a - c)/2)$
7	$\hat{x}_i \triangleq (a + b)/2$

Après cette prédiction, un codage entropique de type Huffman ou arithmétique est utilisé pour effectuer la compression.
Pour plus de détails, se référer à l'annexe H du standard.

Ce modèle de compression très simple n'offre pas des taux de compression fantastiques. La demande croissante pour un standard de compression sans perte à conduit à la normalisation de JPEG-LS, qui permet également de faire une compression quasi sans perte.

4.1.3 LOCO-I

Le standard JPEG-LS (ISO/IEC International Standard 14495-1 – ITU-T Recommendation T.87), principalement normalisé pour palier aux faibles performances du JPEG sans perte, s'appuie sur l'algorithme LOCO-I (*LOw COmplexity LOssless COmpression for Images*) [WSS96, WSS00]. Cet algorithme fut retenu pour sa rapidité accompagnée de bonnes performances.

Les techniques employées par cet algorithme le rendent efficace tout en conservant une faible complexité. En effet, il allie une modélisation de contextes à un codage entropique adaptatif de type VLC, pour rester rapide et efficace en s'adaptant aux variations statistiques de l'image. Tout comme JPEG sans perte, c'est un codage prédictif qui s'effectue en une seule passe. Les pixels voisins précédemment encodés sont utilisés pour déterminer un prédicteur adapté. Le résidu de prédiction est compressé grâce à un codeur adaptatif dérivé de Golomb-Rice dont les paramètres sont déterminés à l'aide d'une sélection de contexte également fonction des pixels voisins.

Pour chaque pixel, une prédiction non linéaire s'appuyant sur une détection rudimentaire de contours est tout d'abord effectuée. Un modèle contextuel est ensuite sélectionné à l'aide des valeurs quantifiées des gradients entre les pixels voisins. Ce contexte, est ensuite utilisé pour modéliser de manière adaptative une distribution de probabilités des erreurs de prédictions. Cette modélisation permet de corriger des biais de prédiction (erreur moyenne ayant tendance à se produire dans une configuration de prédiction particulière), et permet d'estimer efficacement les paramètres d'un codeur Golomb-Rice. Enfin, pour palier au problème de redondance d'un codage VLC qui nécessite au minimum un bit par symbole, un codage RLE est utilisé dans les zones uniformes. Plus précisément, lorsqu'une zone uniforme est détectée, la longueur de la séquence du symbole précédent est encodée, elle aussi sous la forme d'un code Golomb-Rice, dont les paramètres sont estimés à l'aide de la fréquence d'apparition des longueurs des séquences RLE précédentes.

Si x_i est le symbole à encoder, LOCO utilise le motif suivant :

	c	a	d
e	b	x_i	

La valeur de prédiction est : $\hat{x}_i \triangleq \begin{cases} \min(a,b) & \text{si } c \geq \max(a,b), \\ \max(a,b) & \text{si } c \leq \min(a,b), \\ a+b-c & \text{sinon.} \end{cases}$

Et le contexte conditionnant l'encodage de l'erreur est déterminé par les gradients locaux :

$$\begin{aligned} g_1 &= d - a \\ g_2 &= a - c \\ g_3 &= c - b \\ g_4 &= b - e \end{aligned}$$

Dans l'article où LOCO-I fut présenté [WSS96], l'implémentation utilisait 1094 contextes pour des images en niveaux de gris sur 8bits. Le contexte étant sélectionné par quantification des valeurs g_1, g_2, g_3 et g_4.

Le codage s'effectue alors comme suit : en supposant que pour chaque contexte les valeurs de ϵ suivent une distribution Laplacienne centrée en 0, et afin de se rapprocher d'une entrée suivant une distribution géométrique à valeurs entières, les résidus de prédiction $-\alpha/2 \leq \epsilon \leq \alpha/2$ sont associés aux valeurs $0 \leq M(\epsilon) \leq \alpha - 1$ telles que :

$$M(\epsilon) = \begin{cases} 2\epsilon & \epsilon \geq 0, \\ 2|\epsilon| - 1 & \epsilon < 0. \end{cases}$$

Ils est ainsi possible de les encoder à l'aide d'un codeur Golomb-Rice adapté. Le paramètre k du codeur (cf. section 2.2.2) est estimé et mis à jour régulièrement pour chacun des contextes afin d'améliorer l'adaptation aux variations locales des images. Enfin, pour corriger un biais introduit par $M(\epsilon)$ qui a tendance à générer un codage plus court pour les valeurs négatives que pour les valeurs positives, $M(-1-\epsilon)$ est encodé lorsque le centre de la distribution des bits est plus proche de -1 que de 0.

JPEG-LS propose également un mode quasi sans perte (*Near-lossless*) assurant de ne pas provoquer une erreur supérieure à un certain seuil δ pour chaque pixel. Une quantification uniforme est alors appliquée sur le résidus :

$$Q(\epsilon) = \text{sign}(\epsilon) \left\lfloor \frac{|\epsilon| + \delta}{2\delta + 1} \right\rfloor$$

et la prédiction est faite à partir des pixels de l'image reconstruite, et non ceux de l'originale. Ce mode favorise le codage RLE et a donc tendance à introduire des artefacts du type trainées homogènes qui apparaissent lorsque δ est élevé.

4.1.4 CALIC

CALIC (*Context Based, Adaptive, Lossless Image Coding*) [WM97] pousse encore plus loin les principes de contextes et d'adaptabilité. Son schéma de prédiction en est légèrement plus complexe : une sélection de contexte est tout d'abord effectuée à l'aide de quelques pixels déjà connus, une prédiction adaptative utilisant une modélisation des erreurs précédemment commises permet de décorréler l'information et un codeur arithmétique adaptatif est ensuite utilisé pour la compression. CALIC propose également un mode binaire pour les zones de l'image où il ne détecte que deux couleur distinctes, ce mode n'utilise qu'un codage arithmétique.

Dans le mode non binaire, si x_i est le symbole à encoder, CALIC utilise le motif suivant :

		nn	nne
	nw	n	ne
ww	w	x_i	

avec n (*north*), w (*west*), ne (*northeast*), nw (*northwest*), nn (*north-north*), ww (*west-west*) et nne (*north-northeast*).

Le gradient local en x_i est estimé horizontalement par $d_h = |w - ww| + |n - nw| + |n - ne|$, et verticalement par $d_v = |w - nw| + |n - nn| + |ne - nne|$, et est utilisé pour détecter l'orientation et l'amplitude des contours afin d'effectuer une prédiction de x_i adéquat (GAP : *Gradient-Adjusted Prediction*) :

$$\hat{x}_i \triangleq \begin{cases} w & \text{si } d_h - d_v > \alpha, & \% \textit{ contour horizontal très marqué} \\ (X + w)/2 & \text{si } \alpha \geq d_h - d_v > \beta, & \% \textit{ contour horizontal} \\ (3X + w)/4 & \text{si } \beta \geq d_h - d_v > \gamma, & \% \textit{ contour horizontal peu marqué} \\ n & \text{si } d_h - d_v < -\alpha, & \% \textit{ contour vertical très marqué} \\ (X + n)/2 & \text{si } -\alpha \leq d_h - d_v < -\beta, & \% \textit{ contour vertical} \\ (3X + n)/4 & \text{si } -\beta \leq d_h - d_v < -\gamma, & \% \textit{ contour vertical peu marqué} \\ X & \text{sinon.} \end{cases} \quad (4.3)$$

avec $X = (w + n)/2 + (ne - nw)/4$, et $\alpha > \beta > \gamma$ trois constantes fixées respectivement à 80, 32 et 8 dans la publication [WM97] pour des images 8 bits. Pour les images avec une précision plus importante, les valeurs de d_h et d_v sont réajustées pour pouvoir être utilisées dans cette équation.

Cette prédiction $\hat{x}_i$ est ensuite corrigée (correction des biais de prédiction) en fonction d'un modèle d'erreur adaptatif sélectionné par une classification du motif texturel présent sur les pixels voisins (n, w, nw, ne, nn et ww) et par l'erreur ϵ_{i-1} commise sur le pixel précédent : $\hat{x}_i \leftarrow \hat{x}_i + f(n, w, nw, ne, nn, ww, \epsilon_{i-1})$.

L'erreur $\epsilon_i = x_i - \hat{x}_i$ est alors compressée par un codage arithmétique dont les paramètres sont estimés de manière adaptative, conditionnellement au contexte de prédiction/correction.

De par sa conception plus complexe et par l'utilisation du codage arithmétique, CALIC demande plus de calculs que LOCO-I qui fut préféré pour la normalisation de JPEG-LS pour son rapport taux de compression/rapidité. CALIC reste cependant une référence, offrant des ratios de compression meilleurs (cf. chapitre 6).

4.1.5 Autres

Les approches précédentes traitent les pixels de manière séquentielle : ligne par ligne, colonne par colonne (RSO : *Row Scan Order*). De nombreuses études ont été effectuées autour de ce modèle de prédiction. Les algorithmes les plus performants utilisent une modélisation statistique contextuelle, comme LOCO-I et CALIC. Bien que ce dernier soit toujours une référence, quelques algorithmes sont apparus depuis et permettent une compression plus efficace. La plupart de ces algorithmes utilise la méthode des moindres carrés pour construire des prédicteurs adaptatifs [LO01, MT01, YDD03, KL05] mais nécessite plus de calculs.

Ulacha et Stasinski ont proposé un modèle de prédiction basé sur la texture [US07] qui apporte une amélioration non négligeable de l'entropie (d'ordre 0) de l'erreur de prédiction pour des images texturées (-0.2 bpp en comparaison au prédicteur de LOCO-I sur des images de référence). Ils proposent également un codeur contextuel relativement simple [US08] qui semble prometteur. Celui-ci offre des résultats légèrement meilleurs que CALIC (de l'ordre de 0.1 bpp sur des images de référence) pour une complexité similaire.

Une autre catégorie d'algorithmes RSO a pu également émerger. Ceux-ci s'effectuent en plusieurs passes et sont donc beaucoup plus gourmands en temps de calcul.

L'algorithme TMW [MT97, MT98], par exemple, effectue une première passe pour analyser l'image et générer les paramètres d'un modèle statistique. Ce modèle est ensuite utilisé pour effectuer un codage prédictif plus optimal lors d'une seconde passe. Les paramètres du modèle sont d'abord transmis, suivis par l'encodage de l'erreur de prédiction. Cette technique permet d'obtenir des résultats meilleurs que CALIC (entre -0.1 bpp et -0.2 bpp sur les résultats présentés).

FMP [AAB02] obtient des taux de compression similaires à l'aide d'une méthode complexe basée sur la logique floue. Celle-ci nécessite une première étape d'apprentissage qui permet d'optimiser un nombre fixe de prédicteurs $\vec{\phi}_m$. Pour chaque pixel devant être estimé, un nouveau prédicteur $\vec{\phi}(n)$ est conçu à l'aide d'une combinaison linéaire des $\vec{\phi}_m$ dont les coefficients sont estimés par la méthode des moindres carrés sur le voisinage causal. Pour cette algorithme, c'est une matrice d'apprentissage (prédicteurs retenus) qui est transmise, suivie par les données prédites.

Enfin, MRP[1] [MMI00, MOUI05] cherche a optimiser des prédicteurs en entropie, au lieu de minimiser l'erreur quadratique moyenne. Cet algorithme et vraiment très coûteux ($\approx$ 2 minutes pour lena), mais obtient quasiment les meilleurs taux de compression.

La plupart des variantes des codeurs prédictifs RSO en une passe ont pour gros avantage d'être efficaces et de nécessiter peu de mémoire pour effectuer les calculs de prédiction. Seules les quelques lignes précédentes de l'image et les modèles des contextes nécessaires à la prédiction et au codage ont besoin d'être stockés. Les algorithmes peu complexes (comme LOCO-I) seront ainsi préférés pour des systèmes embarqués, ou pour une transmission sans perte rapide. Cependant, ce faible coût mémoire est au détriment d'une représentation non progressive.

Afin d'obtenir cette progressivité, des algorithmes prédictifs tels que HINT (Hierarchical INTerpolation) [RVvDP88] sont utilisables. HINT décompose l'image sous une forme pyramidale permettant d'affiner progressivement sa résolution. Cette pyramide est construite par sous-échantillonnages successifs : une moitié des pixels, pouvant être vus comme l'image basse résolution, permet de prédire l'autre moitié, dont le résidu est vus comme les hautes fréquences de l'image haute résolution. Dans son fonctionnement, HINT est assez proche d'une transformée par lifting (cf. section 3.3) mais ne comporte qu'une étape de prédiction par niveau de décomposition. Il reste donc un codage prédictif et non par transformée (pas de filtrage passe bas).

[1] http://itohws03.ee.noda.sut.ac.jp/~matsuda/mrp/

4.2 Codage par transformée

Contrairement à la prédiction qui fonctionne pixel à pixel, la transformée peut être effectué sur un ensemble de pixels (blocs) ou sur l'image complète. Le but d'une telle approche est de décorréler l'information de l'image à l'aide d'une représentation moins corrélée et moins redondante en information. Ces transformées sont pour la majorité des extensions bidimensionnelles de celles présentées dans la section 3.3.

Dans cette section, les principales techniques de compression par transformée seront présentées. Nous passerons rapidement sur la DCT qui est assez connue et pour laquelle les algorithmes sont assez simples et suivent des schémas relativement similaires. Nous détaillerons un peu plus l'utilisation des ondelettes qui ont conduit à une variété importante d'algorithmes de compression originaux.

4.2.1 DCT

La Transformée en Cosinus Discret (DCT-II) est l'une des transformées les plus répandues en compression d'images. Elle est utilisée par le standard JPEG (ISO/IEC International Standard 10918-1 – ITU-T Recommendation T.81). Son utilisation, tout comme la majorité des transformées basées sur le domaine spectrale, résulte de l'observation que les images naturelles tendent à avoir leur énergie concentrée autour des basses fréquences. La DCT a donc tendance à générer des coefficients hautes fréquences de faible amplitude et ainsi à diminuer l'entropie de la source. Cependant, cette transformée à su montrer ses faiblesses au fil du temps. Le principal reproche va aux artefacts générés lors d'une compression avec pertes. De plus, les fonctions sinusoïdales étant à valeurs réelles, elle génère des pertes d'informations numériques. Elle a tout de même été approximée pour la compression sans perte [WDJ06, Abh07] et est utilisée de façon efficace par Wang *et al.* [WWJ+08] en compression progressive *lossy to lossless* d'images naturelles. Sur les images de test, elle offre des résultats très proches voir meilleurs que JPEG 2000.

La DCT d'une base multidimensionnelle $\mathcal{B}$ est simplement définie par l'application successive de la DCT monodimensionnelle le long de chacun des vecteurs directeurs de $\mathcal{B}$. Son utilisation sur des images (ou blocs) consiste donc simplement à effectuer la transformée de toutes des lignes, puis de toutes les colonnes des coefficients précédemment obtenus.

Comme l'énergie des images est le plus souvent concentrée sur les basses fréquences, les algorithmes de compression avec pertes ont tendance à quantifier plus fortement les hautes fréquences, de manière à obtenir un nombre de coefficients nuls plus importants et à réduire l'entropie. La compression avec pertes peut également utiliser un modèle psychovisuel (cf. section 3.2.3) afin de supprimer des composantes fréquentielles non perceptibles par l'œil humain.

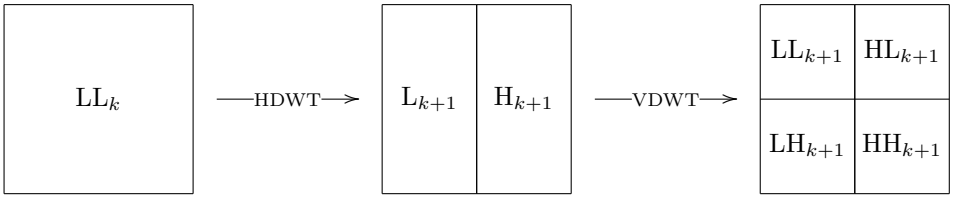
4.2.2 Ondelettes

La transformée phare de ces dernières années pour la compression d'images est la transformée en ondelettes discrète (DWT *Discrete Wavelet Transform*). Elle est notamment utilisée par le standard JPEG 2000. Elle est réputée pour sa simplicité d'utilisation (en particulier grâce au *lifting scheme*), ses bonnes propriétés de décorrélation, sa représentation multi-résolution, etc. Elle a connu un nombre considérable d'applications en compression avec et sans perte d'images. Les ondelettes aidant, la majorité des codeurs qui les utilisent proposent un codage progressif.

La DWT a également inspiré un bon nombre d'autres transformées utilisées en compression ou en débruitage d'images (les paquets d'ondelettes, les curvelettes, les bandelettes, les contourlettes, ...) visant à apporter des propriétés complémentaires (meilleure décorrélation des coefficient hautes fréquences, réduction de l'amplitude des coefficients aux abords des discontinuités/contours, ...).

La DWT bidimensionnelle est souvent effectuée à l'aide de filtres 2D séparables qui permettent une implémentation rapide. Ainsi, pour chaque niveau de décomposition, elle est est appliquée sur les lignes de l'images, générant une représentation basses fréquences et des coefficients hautes fréquences horizontaux. Elle est ensuite de nouveau appliquée sur les colonnes des deux sous-ensembles (sous-bandes) ainsi obtenus. Cette approche est appelée décomposition dyadique ou décomposition en bandes par octave.

Elle peut être modélisée par le schéma suivant :



(4.4)

où HDWT et VDWT sont respectivement la décomposition en ondelettes horizontale et verticale, LL_k la représentation basse résolution de l'image lors de l'étape précédente (ou l'image complète pour LL_0), L_{k+1} celle de basse résolution et H_{k+1} les coefficients hautes fréquences après filtrage horizontal. LL_{k+1} correspond alors à la version basse résolution de LL_k, LH_{k+1} aux coefficients hautes fréquences verticaux, HL_{k+1} à ceux horizontaux et HH_{k+1} aux diagonaux.

Cette décomposition est la plus courante mais il en existe d'autres utilisant des filtres pouvant être basés sur des grilles d'échantillonnage différentes (quinconces, hexagonales, ...) et/ou non séparables. Elles sont la plupart du temps implémentées par schémas de lifting.

Introduit par Shapiro, EZW [Sha93] (*Embedded Zerotree Wavelet*) est l'un des premiers codeurs basés ondelettes à se démarquer. Il fait parti des références de la compression moderne et marque le début de toute une génération d'algorithmes qui se sont inspirés de son concept. En particulier, SPIHT [SP96b] (Set Partitioning in Hierarchical Trees) de Said et Pearlman, qui est encore très utilisé comme référence pour l'évaluation d'autres méthodes et comme base à d'autres algorithmes. Ces deux techniques effectuent un codage progressif de manière à optimiser le rapport débit/distorsion de l'image reconstruite quel que soit le point d'arrêt dans flux de données. EZW et SPIHT sous leur forme originelle tendent ainsi à minimiser l'erreur quadratique moyenne.

Dans cette section seront présentés quelques algorithmes, pouvant être classés en deux grandes familles d'approches (inter- et intra-bande) et une troisième plus petite (approche mixte). Elles se distinguent par les dépendances exploitées pour effectuer la compression des coefficients de la DWT. Ces algorithmes sont principalement conçus pour la décomposition dyadique, mais peuvent être utilisés (ou étendus) avec d'autres décompositions en sous-bandes (DWT quinconce, DCT par blocs, ...).

Approche Inter-Bande

Cette première génération d'algorithmes essaie d'améliorer la compression en utilisant la redondance d'information entre les coefficients d'ondelettes au travers des différentes échelles. Ils exploitent également la propriété croisée des images à contenir plus d'informations dans les basses fréquences, et celle des ondelettes à compacter cette énergie des basses fréquences. Ainsi l'énergie des coefficients hautes fréquences d'une certaine région spatiale a tendance à décroitre avec l'échelle à laquelle est associée la sous-bandes.

EZW EZW [Sha93] effectue une succession d'approximations par quantifications imbriquées afin de permettre un codage progressif (cette quantification est simplifiable en ne considérant que les différents plans de bits). L'information est ordonnée par précision, amplitude, échelle et localisation spatiale, et est embarquée dans une Zerotree spécifiant la carte de signifiance des coefficients. Le Zerotree représente l'arbre hiérarchique entre un coefficient d'ondelette et les coefficients correspondant à la même zone de l'image dans la sous bande de même orientation fréquentielle du niveau de résolution supérieur. Un nœud du Zerotree permet d'indiquer si tous les coefficients qui appartiennent au sous arbre sont inférieurs au seuil de quantification, et le cas échéant, évite de les coder. Afin d'effectuer un codage progressif, ce Zerotree est généré pour un certain seuil de quantification (plan de bits de poids fort) et est raffiné pour chaque nouveau seuil de quantification plus fin (plans de bits suivants). Le codage se fait donc plan de bit par plan de bit, à l'aide d'un parcours en largeur de l'arbre (i.e. résolution par résolution). Cette représentation effectue à elle seule une compression, et un codage arithmétique n'apporte qu'un faible gain sur la progressivité.

SPIHT Tout comme EZW, SPIHT [SP96b, SP96a] utilise les similarités entre les coefficients d'ondelettes des différentes échelles. Cependant l'organisation en arbre hiérarchique des coefficients est utilisée pour effectuer un tri partiel des coefficients, de manière à coder en priorité ceux de forte énergie. Ainsi, le flux de sortie contient une information générée par l'algorithme de tri, nécessaire au décodeur pour pouvoir réorganiser (dé-trier) les coefficients, ainsi qu'une information relative aux valeurs des coefficients (signe et bits de poids faible).

Comme EZW, l'algorithme suppose la décroissance d'énergie des coefficients pour les plus hautes résolutions/fréquences dans l'arbre hiérarchique. Ainsi des coefficients non significatifs dans les basses fréquences ont de fortes chances d'avoir des enfants non significatifs dans les hautes fréquences, ceux possédant une énergie importante ont généralement des enfants possédant moins d'énergie. Ces hypothèses sont utilisées par SPIHT afin de réduire le débit binaire du tri. Le tri partiel réorganise ainsi les coefficients d'une branche en fonction de leur bit de poids le plus fort (réduction de l'erreur quadratique) à l'aide d'un parcours en profondeur des nœuds possédant un descendant significatif dans l'arbre hiérarchique (contrairement à EZW qui effectue un parcours en largeur).

Successeurs de SPIHT SPIHT a connu un grand succès dans le domaine de la recherche, notamment en imagerie médicale. Il a été étendu :

- pour la compression d'images médicales volumiques, par Xiong, Wu, Yun et Pearlman [XWYP98] et amélioré par Cho, Kim et Pearlman dans [CKP04].
- pour le codage de région d'intérêt, par Abu-Hajar et Sankar [AHS02, AHS04] ;
- pour la télémédecine afin de pouvoir transmettre des images à différentes résolutions (ce que ne permet pas SPIHT) et qualités, par Hwang, Chine et Li [HCL03] ;
- et pour la compression d'image médicale à l'aide d'un modèle psychovisuel humain par Prabhakar et Reddy [PR07].

Autres Cho, Pearlman et Said présentent PROGRESS dans [CPS05] qui est un algorithme visant à être léger en temps de calcul lors de la décompression. Pour cela, ils continuent d'exploiter les propriétés inter-bandes, mais abandonnent la progressivité en qualité pour se concentrer sur la progressivité en résolution uniquement. Ils éliminent ainsi le codage par plan de bits, qui devient vite coûteux, pour passer à un codage entropique moins conventionnel. L'arbre hiérarchique permet désormais de modéliser les plages d'amplitudes ($[0]$, $[-1;1]$, $[-3;3]$, ...) auxquelles appartiennent un ensemble de coefficients, de façon à réduire le nombre de bits nécessaires à leur représentation binaire. Les résultats en terme de compression sont assez similaires à SPIHT pour une rapidité de décompression très supérieure.

Approche Intra-Bande

Cette seconde génération ne prend pas en compte la redondance inter-bande et tente d'optimiser la compression des différentes sous-bandes indépendamment. Dans [MCC99] Munteanu *et al.* démontrent qu'une représentation des régions de zéros nécessite moins de symboles avec une représentation par blocs de taille fixe qu'avec une représentation en Zerotree et est donc plus adaptée pour le codage progressif des coefficients d'ondelettes. Bien que cette approche puisse sembler plus efficace pour une compression avec pertes, leur algorithme permet également d'obtenir des taux meilleurs que SPIHT (-0.2 bpp) et similaires à CALIC (+0.02 bpp) en compression sans perte d'images angiographiques.

EBCOT (JPEG 2000) Le standard JPEG 2000 (ISO/IEC International Standard 15444-1 – ITU-T Recommendations T.800) et son extensions (Part 2 : ISO/IEC International Standard 15444-2 – ITU-T Recommendations T.801) sont conçus pour répondre à plusieurs demandes : permettre l'accès aléatoire, la progressivité (spatiale et/ou en qualité) et la possibilité de privilégier une région d'intérêt, en compression avec et sans perte. Il repose sur l'algorithme EBCOT de Taubman [Tau00]. Il n'est pas nécessairement optimal si on se focalise uniquement sur les taux de compression, en particulier à cause de toutes les propriétés auxquelles ils doit répondre. Il offre des taux légèrement inférieurs à JPEG-LS en sans perte, et une courbe débit distorsion intéressante en comparaison à JPEG. Cependant sa complexité calculatoire est assez importante.

Le codeur repose sur une décomposition en ondelettes (éventuellement par blocs (*tile*) pour des images de très grandes tailles). Chacune des sous-bandes est découpée en petits blocs (*Code Block*) typiquement de taille 64×64, et chacun de ces petits blocs est compressé indépendamment.

La compression d'un bloc est alors organisée par plans de bits en 3 passes. La passe de signifiance, celle de raffinement et celle de nettoyage. La passe de signifiance consiste à prédire les coefficients qui devraient devenir significatifs, à transmettre si oui ou non ils le sont et le cas échéant transmettre également leur signe. Un coefficient étant considéré comme significatif si son bit de poids fort est dans le plan de bits courant. La seconde passe consiste à transmettre les bits du plans courant appartenant à des coefficient déjà significatifs lors du traitement du plan précédent. Enfin l'étape de nettoyage consiste à transmettre l'information de localisation des coefficients prédits comme restant non significatifs, mais le devenant, ainsi que leur signe.

Les deux premières étapes sont facultativement compressées à l'aide d'un codeur arithmétique adaptatif contextuel, tandis que la troisième l'est toujours.

Durant la compression de chacun des blocs, des informations concernant la qualité de reconstruction sont retenues et des points de troncature sont insérés dans le flux binaire. Ainsi lorsqu'un codage progressif est effectué, pour chaque couche de qualité souhaitée, une optimisation débit/distorsion permet de sélectionner dans chaque bloc compressé les portions de son flux à transmettre.

Afin d'inclure une région d'intérêt, les bits des coefficients d'ondelettes correspondants à cette région d'intérêt sont tout simplement décalés vers les bits de poids fort avant le codage, de manière à être compressés en priorité.

Pour le lecteur intéressé, l'article de magazine [SCE01] effectue un survol de l'architecture et des fonctionnalités de JPEG 2000. Le tutoriel de Adams [Ada05], disponible avec l'implémentation libre Jasper[2], décrit plus les techniques d'implémentation des algorithmes. Enfin, le livre de référence de Taubman et Marcellin [TM01] permettra d'approfondir la théorie, le fonctionnement et le standard. Ce livre fournit également le code source de la version 2 de Kakadu[3] ; les versions plus récentes du code source sont payantes, mais une version exécutable reste à disposition pour effectuer des tests.

ASSP/AGP L'algorithme ASSP (*Alphabet and Sample-Set Partitionning*) aussi appelé AGP (*Amplitude and Group Partitionning*), également de Said et Pearlman [SP97], est en quelque sorte à l'origine de PROGRESS. Contrairement à celui-ci, il fonctionne en intra-bande et découpe l'organisation des coefficients à l'aide d'un quadtree de manière à regrouper les coefficients de forte amplitude dans des petits blocs, et ceux de faible amplitude dans des blocs plus grands. Pour chacun de ces blocs, les plages d'amplitudes ($[0]$, $[-1;1]$, $[-3;3]$, ...) auxquelles appartiennent les coefficients sont spécifiées, et servent à réduire le débit du codeur entropique. Cet algorithme n'est donc pas progressif en qualité.

SWEET SWEET [And97] se contente d'effectuer un découpage similaire au Zerotree mais par bloc (sans prendre en considération les autres bandes). Ainsi, un quadtree est généré plan de bit par plan de bit. Lorsqu'un bit significatif est détecté sur un certain niveau, l'arbre est subdivisé, et l'algorithme est appliqué récursivement sur chacun des sous blocs. Contrairement à EZW qui effectue la création de son arbre à l'aide d'un parcours en largeur, SWEET le fait par un parcours en profondeur. Lorsqu'un pixel significatif est identifié, tous les bits nécessaires à sa représentation sont transmis. Il n'est donc pas progressif en qualité, mais est plus rapide. La courbe débit-distorsion de Lena présentée dans l'article est assez similaire à celle de SPIHT.

SPECK SPECK [PINS04] de Pearlman, Islam, Nagaraj et Said, reprend l'idée de partition d'ensemble de valeurs utilisée par SPIHT afin de transmettre les coefficients importants en priorité. Cependant SPECK l'applique uniquement à des blocs de coefficients internes à chaque sous-bande, sans utiliser l'organisation hiérarchique, partitionnant chacune d'entre elle à l'aide d'un quadtree permettant d'identifier les pixels significatifs. Il effectue une transmission par plans de bits de l'ensemble des sous bandes et reste ainsi progressif en précision tout comme EZW ou SPIHT. Un ordonnancement des blocs non significatifs de l'ensemble des quadtrees est effectué selon leur taille (les plus petits blocs d'abord). Leur traitement favorise ainsi le codage des coefficients proches de ceux précédemment identifiés comme significatif (et ayant donc une probabilité plus importante de le devenir à leur tour). Ceci permet de favoriser les coefficients de forte amplitude et ainsi de réduire les distorsions pour un débit donné.

Approche Mixte

Cette troisième génération tente d'exploiter les propriétés des deux précédentes, à savoir la localisation spatiale et hiérarchique des coefficients de forte amplitude, mais n'a apporté qu'un faible gain de compression. On peut citer les algorithmes EZBC (*Embedded ZeroBlocks Coding*) [Hsi01, HW02] et WBTC (*Wavelet zero-Block-zero-Tree Coding*) [MK06].

[2] http://www.ece.uvic.ca/~mdadams/jasper/
[3] http://www.kakadusoftware.com/

Conclusion

Ce chapitre aura dressé un état de l'art des techniques de compression les plus connues dans le domaine de la compression sans perte. Il aura permis d'appréhender les différentes méthodes les plus efficaces et la façon dont elles mettent en application les concepts présentés dans les chapitres précédents.

Les techniques prédictives sont souvent celles qui permettent d'obtenir les meilleurs taux de compression, mais elles proposent rarement des solutions au codage progressif qui permet de fournir rapidement un résumé de l'information. Sur ce point les codeur les plus efficaces utilisent presque tous les transformées en ondelettes ainsi qu'un codage par plan de bits, afin de fournir une progressivité sur la qualité de l'image reconstruite. Les approches intra-bande sont actuellement les plus réputées, mais des codeurs inter-bandes très efficaces tels que SPIHT sont toujours d'actualité.

Nous verrons dans le chapitre suivant que la majorité de ces algorithmes sans perte, le plus souvent destinés à des images naturelles, offrent également de bons résultats en compression d'images médicales. Nous verrons aussi comment ils sont étendus pour les images volumiques.

Chapitre 5

Compression d'images médicales

Introduction

Ce chapitre survolera les solutions mises en place pour la compression d'images médicales. Dans un premier temps, nous verrons qu'il existe peu d'algorithmes spécifiques aux images médicales bidimensionnelles, surtout en compression sans perte. Ceux présentés au chapitre précédent, et plus particulièrement les standards, sont donc directement utilisés. Dans un second temps, nous verrons comment ces techniques ont été étendues en compression volumique et comment les codecs cherchent à fournir les propriétés d'accès aléatoire et de codage objet.

5.1 Bidimensionnelles

Les images médicales bidimensionnelles, et surtout les radiographies sont très employées. Les plus anciennes étaient acquises directement sur film et furent ensuite disponibles sous la forme de données informatiques par numérisation de la copie physique. Elle peuvent désormais être acquises à l'aide de capteurs électroniques permettant une quantification numérique immédiate (CR : *Computed Radiography*). Leur quantité importante et leurs dimensions (pouvant par exemple dépasser 2000×3000 sur 16 bits pour une radio des poumons) rendent l'utilisation de la compression intéressante pour leur transfert et leur archivage.

Des études ont été menées afin de comparer les diverses techniques existantes, aussi bien en compression de données qu'en compression d'images [DvAPL97, KOK+98, Clu00]. Ces travaux comparent les résultats d'algorithmes de compression généralistes tels que les commandes unix *pack*(huffman adaptatif), *compress* (basé sur LZW), *gzip* (basé sur LZ77) ou encore le codeur arithmétique adaptatif STAT (de F. Bellard) ; et des algorithmes dédiés à l'image tels que le JPEG sans perte[1], JPEG-LS[1], JPEG2000[1], CALIC[1], SPIHT[1], TMW[1], S+P [SP93, SP96a], etc. Parmi ces codeurs, on trouve une variété d'algorithmes prédictifs et par transformées, avec codage entropique adaptatif (arithmétique, Golomb-Rice) ou non adaptatif (Huffman).

Les résultats de ces études, mettent en évidence que les techniques de compression d'images restent plus appropriées que les méthodes généralistes. CALIC ressort comme le codeur le plus performant sur l'ensemble des images. Certains algorithmes peuvent cependant être plus intéressant selon les modalités : TMW est légèrement plus efficace que CALIC sur les images tomographiques (mais reste beaucoup plus lent), SPIHT est mieux adapté pour les IRM, et JPEG-LS pour les scintigraphies. Leurs taux de compressions sont compris en moyenne entre 2:1 et 5:1. Ces résultats sont similaires à ceux obtenus lors de nos propres expérimentations (cf. chapitre 6).

Ainsi, les standards présentés au chapitre précédent sont couramment utilisés pour la compression sans perte de ces images 2D, de même que pour la compression coupe par coupe des images volumiques. On trouve peu de propositions d'algorithmes sans perte cherchant à s'adapter au contenu afin d'améliorer les résultats sur les images médicales 2D. Cependant des recherches se sont plutôt penchées sur la compression avec pertes. Certains travaux ont cherché à quantifier les taux de compressions acceptables pour les standards actuels et ont contribué à l'élaboration de normes telles que celle de l'ACR (Association Canadienne de Radiologie) [CAR08]. D'autres ont cherché a réduire les distorsions de manière à moins pénaliser le diagnostique. On trouve notamment des travaux sur l'utilisation de la quantification vectorielle [Gau06, GM09], ou encore sur la compression des coefficients de la transformée en cosinus discret de la totalité de l'image (*Full-Frame* DCT) après une quantification adaptée. Cette approche permet de supprimer les effets de blocs indésirables. Certaines techniques plus spécifiques aux modalités à traiter

[1] cf. chapitre 4

ont également été étudiées en mammographie [PPT+03] et en échographie [GSP05]. La première utilise une modélisation de la région d'intérêt des mammographies et une compression adaptée, la seconde tient compte du type de bruit présent dans les échographies et tente de réduire son influence à l'aide d'une quantification adaptée.

5.1.1 Régions d'intérêt

En imagerie médicale, les régions d'intérêt sont assez utiles. Elles permettent en particulier le différencier le fond (inutile au diagnostique) de la forme (données intéressantes). Ainsi le fond peut être compressé avec beaucoup de pertes (voir supprimé) tandis que la forme est compressée sans perte ou avec de faibles dégradations. Dans [PPT+03] le tissu, le contour et le fond des mammographies sont ainsi compressés de manière indépendante. Dans un cadre général, il est possible de spécifier les régions d'intérêt manuellement ou de façon automatique par segmentation/classification.

La transformée en ondelettes a été adaptée pour être appliquée sur des régions de l'image de formes arbitraires comme dans les travaux de Li et Li [LL00] (SA-DWT : *Shape-Adaptive DWT*) et étendue aux ondelettes entières par Abu-Hajar et Sankar [AHS02] (ISA-DWT : *Integer SA-DWT*) qui l'utilisent avec un codeur dérivé de SPIHT pour obtenir de bonnes performances sur des images naturelles [AHS04].

5.1.2 Progressivité

La majorité des algorithmes de compression progressive employés en image médicales se basent sur des décompositions ondelettes (SPIHT, EBCOT, ...). Dans [GEVK00] Grüter *et al.* proposent également l'utilisation de la ROPD (*Rank-Order Polynomial Decomposition*) pour la compression progressive de ces images. Elle s'appuie sur une décomposition en sous-bandes morphologiques [ELK95], à l'aide de prédicteurs polynomiaux non linéaires.

Le codeur LAR (*Locallly Adaptive Resolution*) [DBBR07] initialement conçu pour effectuer une compression avec pertes scalable a également été étendu pour effectuer une compression *lossy-to-lossless* de manière efficace [BDR05, DBM06, PBDB08]. Il repose sur une représentation de l'image sous la forme d'une partition en quadtree de zone relativement homogènes auxquelles est associé un niveau de gris représentatif. Une image grossière est construite par interpolation dans les zones homogènes, et est utilisée pour prédire l'originale. L'erreur d'approximation est considérée comme la texture et est codée de manière multi-résolution en s'appuyant sur la décomposition en quadtree précédente. Ce modèle de représentation simple peut être utilisé et étendu pour satisfaire divers critères, tels que la progressivité ou le codage de ROI, et diverses techniques de décorrélation de l'information texturelle peuvent être mises en place selon le contexte d'utilisation.

5.2 Volumiques

L'extension la plus triviale de la compression d'images 2D pour des images volumiques est l'encodage de chacune des coupes indépendamment des autres. Cette technique est très clairement sous-optimale, en terme de taux de compression, puisqu'elle ne prend pas en considération les corrélations pouvant exister entre des coupes successives. Cependant, elle est couramment employée en compression de vidéos (considérées également comme des images volumiques), lors de l'acquisition, afin de réduire la complexité des algorithmes, de conserver une bonne qualité ou de faciliter le montage en studios par exemple. En imagerie médicale c'est une approche qui peut être employée lorsque le contexte d'utilisation nécessite de favoriser l'accès aléatoire à des coupes éparses.

Afin de supprimer la redondance d'information entre les images successives les techniques de décorrélation prédictives et par transformées sont également applicables.

Pour des approches prédictives, si on considère le volume comme une séquence d'images, une coupe particulière peut être prédite à l'aide des coupes précédemment encodées. Dès lors, seule l'erreur de prédiction nécessite d'être compressée et transmise. Afin d'améliorer cette décorrélation, on peut également transmettre une information supplémentaire spécifiant des actions à effectuer (une compensation du mouvement par exemple). Il faut alors souvent faire appel à des techniques d'optimisation pour minimiser conjointement la quantité d'information supplémentaire et la quantité d'information décorrélée.

Pour les approches par transformées, des extensions tridimensionnels des techniques employées sur les signaux 1D et les images 2D peuvent être utilisées.

Dans cette section sera effectué un état de l'art, plutôt orienté imagerie médicale, des techniques de décorrélation volumique en commençant par les approches prédictives, très utilisées en vidéo, puis en

enchaînant sur celles par transformées, plus utilisées en imagerie médicale et satellitaire et commençant également à émerger en compression vidéo très scalable. Nous conclurons sur l'application de ces outils en milieu médical. Les termes « coupe » et « image » seront indifféremment employés par la suite.

Le lecteur pourra également se référer aux états de l'art [PvAdRD01] et [SMB+03] qui sont assez complémentaires, le premier étant plus orienté sur le codage prédictif, et le second sur le codage par transformées.

5.2.1 Codage prédictif

Le domaine de la vidéo (qui peut être considéré comme une modalité volumique) et plus précisément celui de sa compression possède une vaste littérature dans laquelle le codage prédictif est très apprécié. Certaines techniques ont été améliorées durant de nombreuses années et sont ainsi devenues très efficaces. Elles sont parfois utilisées en compression avec pertes d'images médicales volumiques.

Les codecs vidéos "grand public" sont conçus de façon à offrir un décodage qui puisse être effectué en temps réel. Ils essaient de fournir un accès aléatoire assez fin pour faciliter la navigation et permettre un positionnement temporel dans la vidéo qui soit rapide. Les plus récents cherchent également à être très progressifs afin d'offrir la possibilité d'effectuer du streaming pour différents débits et/ou résolutions spatiales et/ou résolutions temporelles. Pour l'accès aléatoire, la technique du GOP (*Group Of Pictures*) est très employée. Elle consiste à compresser les séquences par paquets d'images de longueurs fixes (ex : 8 ou 16 images). Pour de tels codecs, le codage prend souvent beaucoup plus de temps de façon à mieux décorréler l'information et à réduire les calculs lors du décodage pour permettre une reconstruction et un affichage temps réel.

On dissocie trois approches classiques de prédiction volumique. La première consiste à prédire la coupe à compresser à partir d'autres coupes déjà connues (les précédentes par exemple). L'image de l'erreur de prédiction volumique est ensuite décorrélée à l'aide de techniques 2D. Ceci s'effectue souvent à l'aide de transformées (DCT-2D ou DWT-2D). Les deux approches suivantes sont des extensions des techniques bidimensionnelles. Elles utilisent l'information volumique disponible en chacun des pixels à compresser. On distinguera alors l'extension volumique séquentielle (DPCM), qui se résume en un parcours coupe par coupe, ligne par ligne, et une prédiction pixel par pixel ; de l'extension volumique hiérarchique (HINT) qui utilise une représentation volumique basse résolution (par simple sous-échantillonnage) pour prédire les pixels à une résolution supérieure.

Dans la suite de cette section seront commentées les approches visant à effectuer la prédiction de la totalité d'une image ainsi que les extensions de la prédiction bidimensionnelle (RSO et hiérarchique).

Prédiction de coupe

Le standard de compression vidéo le plus récent, H.264/MPEG-4 AVC (*Advanced Video Coding* : ITU-T Recommendation H.264, ISO/IEC MPEG-4 AVC 14496-10) qui est une évolution des standards H.261, H.262 et H.263 précédents, relève des travaux en commun du groupe H de l'ITU, spécialisé dans l'audiovisuel et les systèmes multimédias, et du groupe de travail MPEG (*Moving Picture Experts Group*) d'ISO/IEC. Sans rentrer dans les détails (le lecteur pourra se référer à [Ric03]), il offre de nombreuses fonctionnalités, dont la scalabilité temporelle, l'accès aléatoire et la décomposition en objets.

L'accès aléatoire est mis en place avec l'aide des GOPs, et la scalabilité temporelle (permettant d'encoder la vidéo à diverses fréquences d'acquisition temporelle dans le même flux de données) est permise grâce à une mise en place astucieuse de divers prédicteurs d'image. On distingue en particulier les images de type I (pour *Intra*), P (pour *Predicted*) et B (pour *Between* ou *Bi-predicted*. Les images I sont compressées en intra (ne nécessitant pas d'informations sur les autres images), comme l'indique leur nom. Les images P sont compressées après une prédiction à l'aide d'image(s) de références précédentes, et les images B après une prédiction à l'aide d'image(s) de références précédentes et/ou suivantes déjà encodées. Cette technique permet également d'obtenir de bons rapports débits/distorsion.

L'image devant être prédite est découpée en blocs de taille fixe. Pour chacun d'entre eux, les images de référence sont utilisées pour rechercher un bloc le plus similaire possible, qui sera employé pour la prédiction. Cette technique permet d'effectuer une compensation du mouvement dans les vidéo, et la localisation du bloc le plus ressemblant est codée à l'aide d'un vecteur souvent appelé vecteur (de compensation) de mouvement. Cette localisation peut être sub-pixelique, au quel cas les blocs prédicteurs sont générés à l'aide d'une interpolation par filtrage.

L'image d'erreur est ensuite compressée avec pertes à l'aide d'une TCD-2D.

La succession de coupes des images volumiques médicales ne respectent pas l'hypothèse d'objet en mouvement par translation. Les contours/surfaces des différents objets volumiques (organes...) représentés coupe-à-coupe ont certes tendance à se déplacer mais également à se déformer/dilater/contracter. Cependant, ces déformations seront localement faibles si la résolution en (z) est assez fine (comme les TDM coupes fines par exemple). Des prédicteurs utilisant une estimation de mouvement permettent donc de décorréler l'information mais seront moins efficaces à plus faible résolution (IRM par exemple).

Ainsi, Nosratina *et al.* [NMOL96], et Srikanth et Ramakrishnan [SR05] effectuent la compression d'IRM à l'aide de modèles de prédiction par compensation de déformations. Ces modèles utilisent les déformations d'un maillage pour prédire des images successives. Bien que ces articles ne proposent pas de résultats pouvant être aisément comparés, on peut supposer que, pour les images ayant une résolution en (z) assez faible, une telle approche puisse obtenir de meilleurs résultats qu'avec une compensation de mouvement classique, par blocs.

Prédiction séquentielle

Proche de la prédiction de coupe, dos Santos et Scharcanski proposent dans [dSS08] une compensation de mouvement en deux étapes pour la compression sans perte et presque sans perte d'angiographies (représentation temporelle). Un *bloc matching* est effectué pour trouver le bloc le plus ressemblant dans une image de référence (image précédente), et est utilisé pour générer un prédicteur linéaire adaptatif du pixel courant. Les résultats expérimentaux de cette technique, de complexité calculatoire importante, n'améliorent que faiblement la compression effectuée frame par frame par JPEG-LS (de l'ordre de 0.08 bits/pixel pour des images 8 bpp, et 0.13 bpp pour des images 12 bpp).

Comme l'estimation du mouvement est très coûteuse en temps de calcul, De Rycke et Philips [dRP99, PvAdRD01] utilisent un codeur intra (JPEG-LS) pour la majorité de l'image et une compensation de mouvement uniquement sur les pixels pouvant être potentiellement mal prédits par le prédicteur 2D. Ce changement de mode est effectué automatiquement, avec l'aide de l'image précédente, et ne nécessite donc d'aucune information supplémentaire. En pratique, moins de 1% des pixels sont prédits par la compensation de mouvement.

Diez-Garcia *et al.* [DGSWAL05] utilisent un prédicteur volumique adaptatif en une passe utilisant un schéma similaire à la prédiction RSO de LOCO ou CALIC étendu sur une troisième dimension. Ainsi, les voxels utilisés pour la prédiction appartiennent à la coupe précédente et à la portion RSO déjà prédite de l'image. Cette approche offre de meilleurs performances que les codeurs par transformée volumique mais, comme pour les schémas prédictifs 2D, ne permet pas d'offrir la progressivité. De plus, la méthode adaptative proposée devient plus efficace sur des séquences longues (aussi bien en temps de calcul qu'en taux de compression), les GOP de petites tailles pour un accès aléatoire rapide ne sont donc pas envisageables. Cette technique est alors plus adaptée pour de l'archivage à long terme, lorsque le volume complet doit être récupéré, ou encore pour une consultation séquentielle.

Prédiction hiérarchique

Roos et Viergever [RV93] utilisent une approche DPCM volumique simple, et la comparent avec un extension 3D de HINT. Leur conclusions sont qu'une extension volumique n'apporte qu'un faible gain de codage. Ces résultats restent en concordance avec l'article de Aiazzi *et al.* [AABA96] qui présentent GRINT (*Genralized Recursive Interpolation*) une généralisation de HINT, dont ils comparent l'entropie d'ordre zéro après décorrélation 2D et 3D. Sur les résultats présentés, seule une faible amélioration (-0.05 bpp) est notée avec le passage du 2D au 3D. Cependant HINT-3D et GRINT peuvent permettre d'effectuer un codage progressif du volume.

5.2.2 Codage par transformée

Comme pour les transformées bidimensionnelles, une simple extension mathématique des transformées monodimensionnelles suffit à obtenir des transformées volumiques. Dans ce cas, les différences viennent le plus souvent des techniques utilisées pour réorganiser les données que de l'extension de la transformée. L'architecture des codeurs est donc sujette à discussions. Ils sont souvent des extensions volumiques des codeurs ayant fait leurs preuves sur les images 2D. Ils reprennent les principes algorithmiques et les étendent sur des volumes de pixels. Ainsi une décomposition en quadtree devient une décomposition en octree ou un codage par "petits" blocs (EBCOT) devient un codage par "petits" cubes. L'un des états de l'art incontournable est celui de Schelkens *et al.* [SMB+03], qui détaille et compare les résultats de quelques algorithmes volumiques.

Pour l'archivage médical, un gain de compression en comparaison aux mêmes algorithmes 2D est généralement notable mais il n'est pas fantastique (cf. chapitre 6). La plupart de ces codeur permettent une représentation progressive de la totalité du volume. Cette fonctionnalité est intéressante lorsque plusieurs qualités différentes sont nécessaires, ou lorsqu'une représentation basse qualité du volume est suffisante pour pouvoir localiser les coupes et/ou objets réellement utiles et devant être transférés sans perte. Malheureusement ces codeurs n'offrent pas souvent la possibilité de récupérer une information localisée.

Des techniques un peu plus complexe d'extension de la transformée ont tout de même fait leur apparition. Il existe notamment une demande croissante pour des codecs très scalables en vidéo, permettant de représenter, dans un même flux binaire, différentes résolutions d'image, différentes résolutions temporelle et ce pour différents débits. Une solution pouvant être adoptée consiste en un compensation du mouvement par blocs, temporellement décorrélés à l'aide d'ondelettes. WAVIX [VGP02, BFG05], et MC-EZBC[2] [HW01, WC02](*Motion-Compensated EZBC*) sont deux exemples suivant cette approche. WAVIX utilise JPEG 2000 pour compresser chacune des images de chacune des sous-bandes décorrélées temporellement ($\approx$ 3D-EBCOT). MC-EZBC a un fonctionnement relativement similaires mais effectue le codage à l'aide de 3D-EZBC.

5.2.3 Accès aléatoire

Certaines techniques de codage prédictif peuvent être très compétitives pour leurs taux de compression, rapidité et coût mémoire pour le codage/décodage. Cependant l'approche prédictive adaptative devient beaucoup plus critique pour l'accès à une coupe particulière. En effet, pour obtenir toute l'information nécessaire à sa décompression (contexte/prédicteur), il faut préalablement transférer et décoder la totalité des informations dont elle dépend. De même des algorithmes comme 3D SPIHT obligent un décodage complet du volume pour pouvoir accéder à une coupe particulière sans aucune pertes.

Une solution pour réduire ce problème est de travailler par groupes de coupes (GOS : *Group Of Slices*), comme en vidéo avec les groupes d'images (GOP : *Group Of Pictures*) de longueurs réduites (par la suite on ne différenciera pas GOS et GOP, de même l'axe transaxial (z) des volumes sera confondu avec l'axe temporel des vidéos). Ainsi on obtient une coupe particulière en ne décodant que l'information nécessaire contenue dans le GOS, au détriment d'une baisse d'efficacité (pouvant être importante dans le cas de [DGSWAL05]). Les GOPs d'un codage prédictif offrent souvent des images codées en intra uniquement (la première image par exemple), tandis que des GOPs résultant d'une transformée en ondelettes offrent des représentations progressives : en résolution temporelle et spatiale ou en qualité. Un résumé du contenu du volume est ainsi disponible pour un faible coût et permet un accès aléatoire plus rapide à une sélection des coupes (ou de plages de coupes) à consulter dans une qualité supérieure. Cette approche peut ainsi permettre de réduire les transferts réseaux et accélérer la visualisation lors d'une interrogation à distance, en comparaison à une compression totalement volumique.

Les GOS peuvent être une solution acceptable si les utilisateurs accèdent en général à des séquences de plusieurs coupes consécutives (cas de la vidéo). Par contre, si ceux-ci ne cherchent à accéder qu'a des coupes très éparses, la taille optimale du GOS risque d'être 1 soit un codage intra uniquement, à moins que la compression d'un GOS de taille k ne permette d'obtenir des taux de compression k fois supérieur, ce qui n'est a priori jamais le cas en compression sans perte. Cependant, l'accès aléatoire à des coupes se fait logiquement à la suite d'une sélection effectuée sur une version dégradée du volume par exemple. Ainsi comme l'information du résumé est déjà à disposition, seule un apport d'information supplémentaire est nécessaire à la reconstruction de la coupe sélectionnée.

Menegaz et Thiran [MT03] effectuent une compression volumique à l'aide d'ondelettes, et organisent le codage de façon à pouvoir favoriser un accès aléatoire par coupe. Cependant la technique proposée n'est pas rentable lorsqu'une seule coupe doit être récupérée. En effet, selon la taille des filtres d'ondelettes utilisés pour la décorrélation inter-coupe, une quantité importante de coefficients doit être récupérée pour pouvoir décoder une seule image. Ce nombre de coefficients croit de manière exponentiel avec le nombre de niveaux de décomposition le long de l'axe (z) et la compression intra de la coupe en question devient rapidement plus rentable. Les résultats présentés pour leur version bidimensionnelle sont tout de même meilleurs que JPEG-LS et leurs algorithmes permettent d'effectuer un codage progressif qui peut s'avérer bénéfique si le sans perte n'est pas nécessaire.

[2] codes source disponibles : `http://www.cipr.rpi.edu/research/mcezbc/`

Dans l'optique d'améliorer l'efficacité de 3D SPIHT, Cho, Kim et Pearlman [CKP04] proposent l'utilisation d'une décomposition en arbre asymétrique pour le trie partiel des coefficients (AT-SPIHT : *Asymetric Tree SPIHT*). Ils montrent qu'il reste efficace même sur des GOPs de petite taille. L'accès aléatoire est ainsi favorisé.

Wu et Qiu [WQ05] ont également cherché un algorithme rapide permettant un accès aléatoire efficace. Leur proposition, M3DW (*Modified 3D dyadic Wavelet*), utilise une décomposition dyadique volumique non conventionnelle. Ils estiment que les coefficients d'ondelettes hautes fréquences ne sont pas suffisamment corrélé entre des coupes successives. La décomposition utilisée est alors une décomposition dyadique 2D classique sur un niveau, suivie d'une décomposition inter-coupe en ondelettes de Haar uniquement sur les basses fréquences. Ce processus peut ensuite être réitéré sur le volume basse résolution. La compression des coefficients est ensuite effectuée avec un codeur Golomb-Rice adaptatif pour rester rapide.

Le choix des ondelettes de Haar pour la décorrélation inter-coupe n'est pas anodin. Celles-ci, ayant le support le plus compacte, permettent de réduire au maximum le nombre de coefficients nécessaires au décodage d'une seule coupe. Avec cette technique, le nombre valeurs non compressées qu'il est nécessaire de transmettre pour la reconstitution d'une coupe est de l'ordre de 4/3 supérieur à un codage intra. Cependant résultats sur la base 8 bits du CIPR (cf. section 1.4) montrent que le gain de codage apporté par cette technique dépasse généralement les 33% en comparaison à JPEG-LS. Ainsi le débit binaire compressé nécessaire au décodage d'une seule coupe est légèrement inférieur à celui de JPEG-LS en intra. Les taux de compression sont également meilleurs que 3D SPIHT (-0.12 bpp) lui même plus performant que CALIC (-0.36 bpp) et permettent un gain moyen supérieur à 0.8 bpp en comparaison à JPEG 2000. Enfin le décodage d'un volume complet est plus rapide qu'avec JPEG-LS (réputé pour sa rapidité). Il en résulte que cette technique rapide est à la fois efficace pour le stockage et pour la transmission de coupes aléatoires.

5.2.4 Objets

Menegaz et Thiran ont proposé un algorithme volumique de compression d'une décomposition de l'image en objets d'intérêts permettant un décodage progressif de chacun d'entre eux, indépendamment [Men00, MT02]. Leur approche repose sur une décomposition en ondelettes classique sur la totalité du volume. Pour chaque région d'intérêt, l'ensemble des coefficients nécessaire à sa reconstruction sont conservés et encodés (dans l'article deux codeurs 3D sont considérés EZW-3D et MLZC). Ainsi, les coefficients d'ondelettes contribuant à plusieurs objets (proches des contours) sont codés de façon redondante. Une alternative à cette redondance serait l'utilisation de la transformée en ondelettes adaptée à une forme arbitraire (cf. section 5.1.1).

Menegaz et Thiran supposent que les délimitations des objets sont déjà connues du codeur et du décodeur, et ne considèrent pas le codage de celles-ci. Durant sa thèse sur les images médicales 2D [Che07], Chen a proposé un découpage polygonal pour leur représentation afin de réduire leur coût en comparaison à la compression d'un masque binaire [CTC06].

Conclusion

Ce chapitre aura présenté quelques techniques de compression employées en imagerie médicale. Pour les images 2D, on trouve peu d'algorithmes spécialisés. En imagerie volumique on trouve une littérature un peu plus riche, qui tente de répondre aux besoins du milieu hospitalier : les images volumiques nécessitent un archivage performant, sans dégradation ou avec des dégradations contrôlées et offrant des possibilités de consultation non pénalisantes pour le réseau qui est de plus en plus sollicité. L'idéal serait un codec offrant un bonne compression, autorisant un accès aléatoire rapide, proposant une qualité progressive et/ou contrôlable et permettant la distinction de régions d'intérêt ; cependant des codecs adaptés à chaque cas d'utilisation seraient plus optimaux.

Les approches 2D ont connu de nombreuses extensions volumiques et quelques travaux se sont déjà penchés sur la problématique d'une compression efficace permettant un accès aléatoire rapide (localement ou via un réseau). Les résultats fournis par ces travaux sont prometteurs. La notion de régions d'intérêt et de codage d'objets peut également être importante et se révéler bénéfique : fixer des qualités adaptées aux besoins, éviter de transférer/stocker des région inutiles. Il existe déjà quelques techniques intéressantes pour leur compression, ainsi que pour représenter leur localisation.

Dans le prochain chapitre seront présentés des résultats de quelques unes de nos expérimentations sur les sujets de la compression par coupe, volumique, avec régions d'intérêt, et presque sans perte.

Chapitre 6

Expérimentations

Introduction

Dans ce chapitre, nous présenterons les résultats et interprétations de différentes expérimentations. Les deux premières sections regroupent les bilans d'études qui ont été menées afin d'évaluer des algorithmes de compression existants en compression bidimensionnelle et volumique, et dans le but d'analyser les avantages et inconvénients de chacun sur des images médicales. Les troisième et quatrième sections décrivent des méthodes de compression avec pertes contrôlées à savoir la compression sans perte de la région d'intérêt (ROI) uniquement, et la compression presque sans perte. Elles contiennent également une analyse de leurs résultats.

6.1 Compression intra-coupe

Les codeurs d'images 2D, appliqués coupe par coupe, sont souvent utilisés pour réduire les coûts de transfert et de stockage des images volumiques médicales, et certains sont même intégrés au standard DICOM (*Digital Imaging and COmmunications in Medicine*). Ils fournissent donc une bonne base de comparaison pour les techniques de compression multi-coupes (volumiques).

JPEG-LS et JPEG2000 sont deux standards performants, CALIC et SPIHT deux codeurs de référence. JPEG-LS et CALIC sont des codeurs de type prédictifs (cf. 4.1), tandis que JPEG2000 et SPIHT utilisent une transformée en ondelettes (cf. 4.2) et proposent un codage progressif.

Le codec de Microsoft : HD-Photo, [TSS+08] en phase de devenir un nouveau standard (JPEG-XR) a également été testé, mais celui ci ne s'est pas avéré d'une grande efficacité pour la compression sans perte. En comparaison à JPEG-LS, les taux de compression vont, selon les séquences, de +0.4 à +1 bpp sur des images 12bits, et peuvent dépasser +2bpp sur les images 16bits. Ces faibles performances ne seront donc pas présentées/commentées.

Les résultats obtenus rejoignent ceux d'études déjà menées [DvAPL97, KOK+98, Clu00]. Quelques courbes sont disponibles en Annexe A. En ne considérant que des images natives, CALIC offre presque toujours les meilleurs taux de compression (il se fait parfois devancer pas SPIHT sur quelques IRMs). Les résultats de JPEG-LS suivent à peu près la même progression que ceux de CALIC avec des pertes de l'ordre de +0.2bpp (ce résultat se vérifie également sur des images retouchées).

Comparativement à ces codeurs prédictifs, les codeurs ondelettes cherchant à être progressifs par raffinement successif sont moins stables. Dans quelques situations SPIHT obtient des résultats similaires à CALIC, mais les pertes vont jusqu'à +0.3bpp et sont en moyenne de l'ordre de +0.15bpp. JPEG2000 semble encore moins stable et fournit un débit supérieur à CALIC de +0.15bpp à +0.4bpp et est presque toujours moins performant que SPIHT. Ces résultats sont dépendants de la nature des images et plus particulièrement du bruit présent (cf. figures 6.1 et 6.2). En effet plus les artefacts de rétroprojection des tomographies (enchevêtrements de droites d'amplitudes variables) sont importants, plus les taux de compression divergent des résultats de CALIC et JPEG-LS. Ce bruit génère beaucoup de hautes fréquences orientées et son amplitude joue un rôle non négligeable sur les taux de compression des codeurs ondelettes. En effet, JPEG2000 et SPIHT utilisent une décomposition dyadique qui implique que 3/4 des coefficients d'ondelettes à compresser se situent dans les bandes de plus hautes fréquences. Le bruit vient donc perturber ces codeurs qui ne favorisent que quelques orientations pour la décorrélation. De plus, les ondelettes entières utilisées ne forment pas des bases orthonormales. Ainsi, l'énergie des coefficients

(A)

(B)

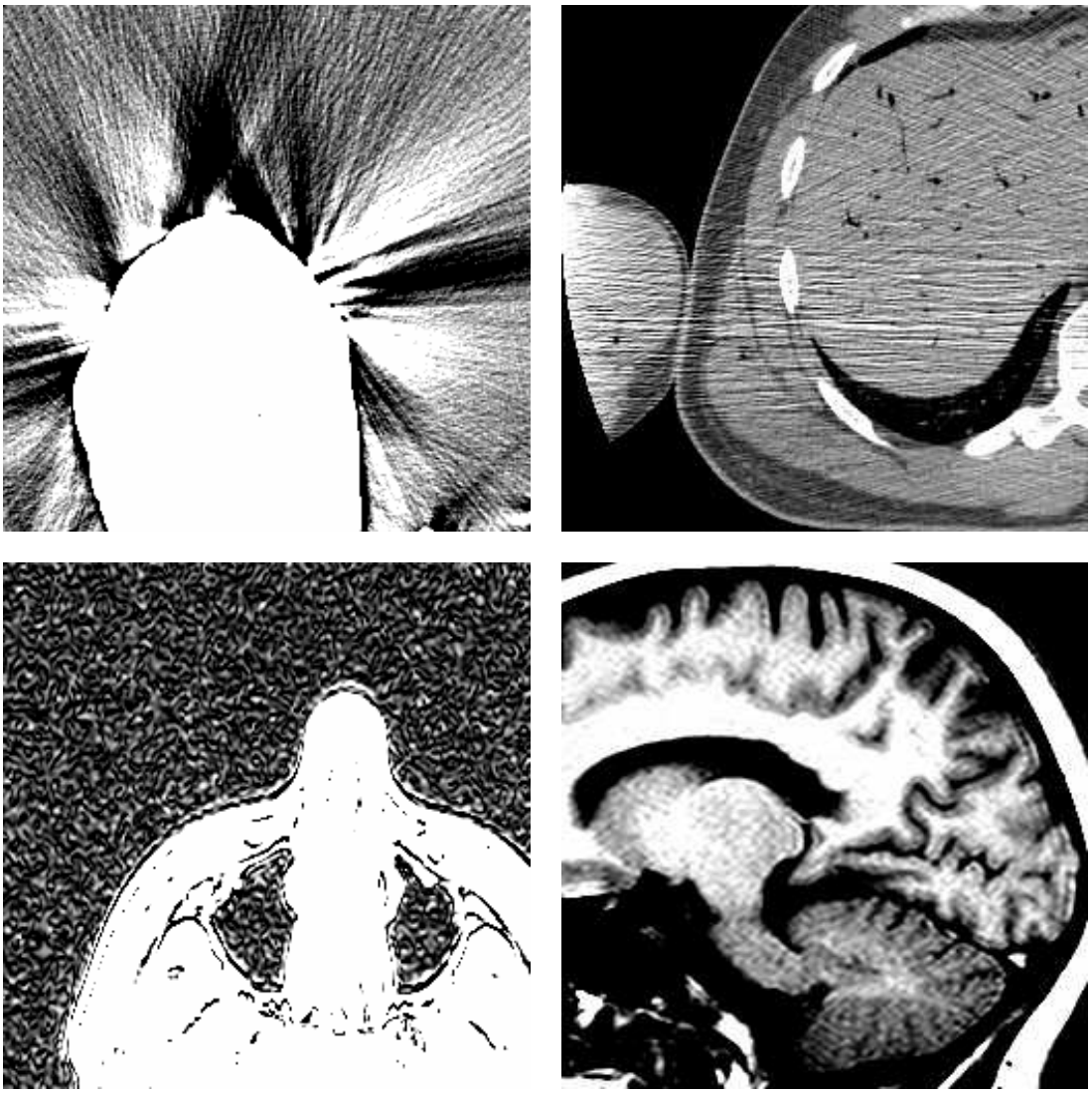


FIG. 6.1 – Illustration du type de bruit présent sur les tomographies (A) et les IRMs (B) pour le fond (colonne 1) et pour la forme (colonne 2). Les images ont été générées en ajustant la plage de niveaux de gris de façon à mettre en évidence le bruit. Comme on peut le constater, le bruit en tomographie peut avoir des amplitudes très variables (colonne 1). Il est très orienté et peut agresser les contours (colonne 2). Par contre les IRMs ont un bruit relativement texturé, plutôt homogène et ne perturbe que légèrement les contours (effet de Gibbs).

hautes fréquences (et donc en particulier tous ceux liés au bruit) après décorrélation est amplifiée, ce qui fait également décroitre les performances de compression.

Cependant, le bruit présent sur les IRM, qui à tendance à comporter moins de hautes fréquences et à être plus structuré spatialement (texturé), favorise légèrement la décorrélation par ondelettes et donc les taux de compression de SPIHT et JPEG2000.

Souvent les images volumiques sont retouchées en vue d'effectuer des modélisations 3D des organes, des muscles, des vaisseaux et/ou de l'ossature. Il a pu être constaté que ces clichés retouchés peuvent être archivés dans les dossiers médicaux. La majorité des traitements effectués consistent à réduire le bruit de façon à pouvoir dissocier plus aisément les différentes régions. Ce filtrage a un impact important sur la compression : comme les hautes fréquences sont supprimées ou réduites (le filtre peut chercher à préserver les contours et ne supprime pas nécessairement tout le bruit) SPIHT et JPEG2000 deviennent plus efficaces. Leurs taux de compression ont tendance à se placer entre ceux de CALIC et ceux de JPEG-LS. Les pertes par rapport à CALIC deviennent donc inférieures à 0.2bpp. Le filtrage a également un impact sur les autres méthodes de décorrélation, les taux de compression sont donc meilleurs.

Cette remarque est également valable pour les images tomographiques reconstruites avec l'aide d'algorithmes statistiques et/ou volumiques, qui prennent plus en compte le bruit d'acquisition.

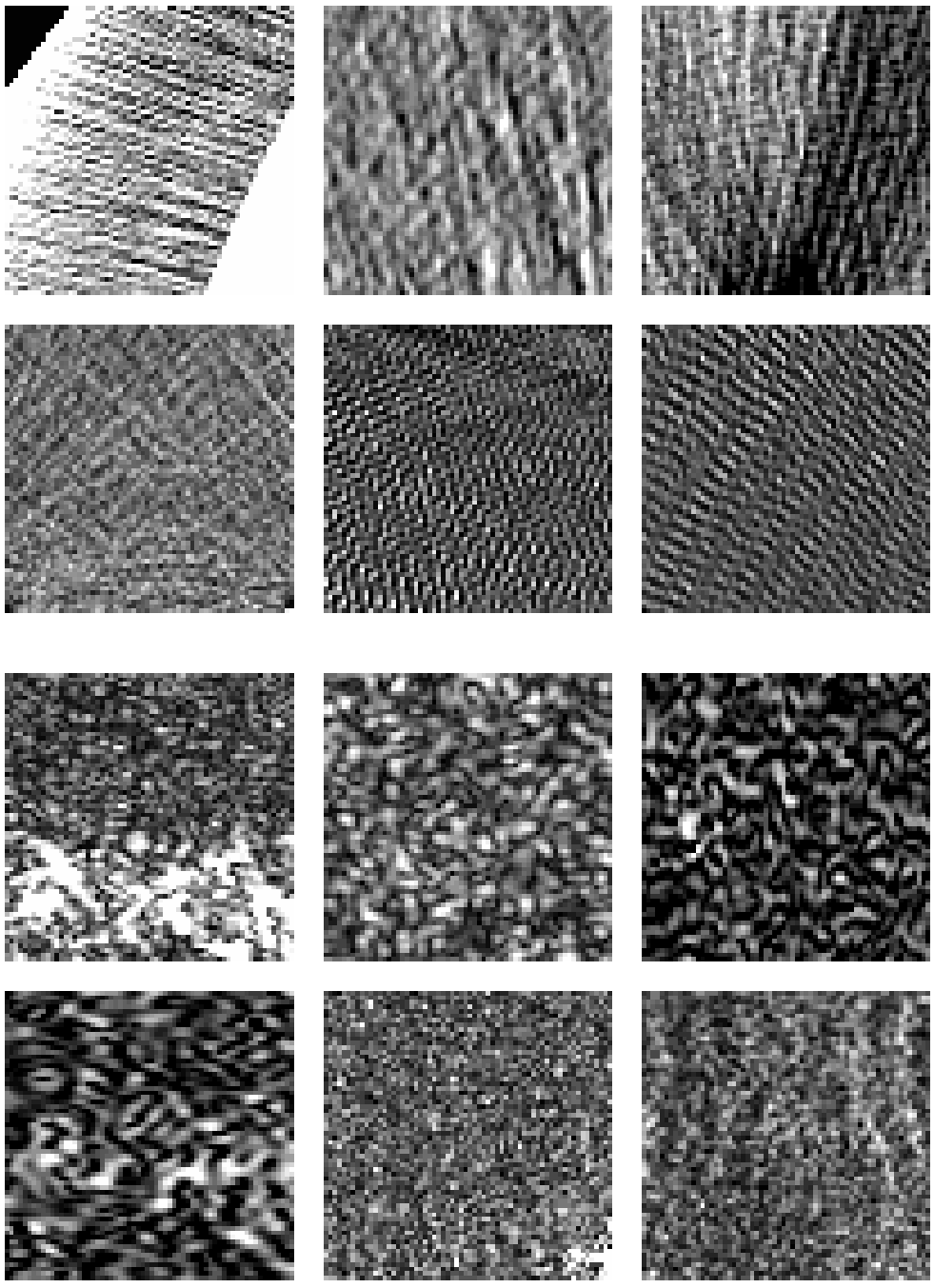


FIG. 6.2 – Patchs illustrant le type de bruit présent sur les tomographies (A) et les IRMs (B). Les patch ont été pris sur le fond des images et ont été générées en ajustant la plage de niveaux de gris de façon à mettre en évidence le bruit.

Ainsi les résultats pratiques rejoignent la théorie (cf. section 3.4.2) : les codeurs prédictifs sont souvent les plus efficaces, mais ne proposent pas de progressivité.

6.1.1 Quelques chiffres

Sur des images en niveaux de gris sur 12 bits (pas toujours tous utilisés) le débit après compression par CALIC varie entre 2bpp pour des images ne contenant quasiment que du bruit d'assez faible amplitude et 6bpp dans des cas extrêmes. Les images contenant peu d'informations (extrémités de crane ou pieds) ont généralement un débit de l'ordre de 3bpp. Pour un crâne, les coupes d'IRMs ont un coût allant de 2.5bpp à 5bpp, celles de scanner sont entre 3bpp et 4.5bpp. Les images les moins compressées sont des scanners natifs bruités de thorax situés entre 4bpp et 6bpp.

Nous ne disposons que d'un dossier médical contenant des images sur 16 bits : les IRM d'un genou. Le débit atteint par CALIC (pris comme référence) est dans le meilleur des cas 6.9bpp (pour une image dont quasiment 1/3 des pixels sont occupés par des bandes noires, générées après reconstruction afin d'obtenir des images de taille carrée (512×512), soit environ 9.9bpp effectifs) et dépasse légèrement les 10bpp dans le pire des cas. Les 8 bits de poids faible des zones de reconstruction de ces images semblent totalement aléatoires : l'entropie de ces plans de bits après une représentation en code Gray (i.e. code binaire réfléchi) est très proche de 1, et aucune corrélation n'est apparente (ce code a tendance à rendre les plans de bits plus corrélés : des valeurs proches ont un nombre de bits différents plus faible qu'avec une représentation classique) ; et l'écart type des coefficients d'ondelettes de la bande hautes fréquences diagonale (HH) (après une décomposition dyadique 5/3) dépasse 220 ce qui est de l'ordre de 10 fois plus que pour une CT native 12 bits.

Contrairement aux images 12bits JPEG-LS offre des débits très similaires à CALIC (inférieurs à +0.05bpp) et donne même des résultats légèrement meilleurs sur quelques clichés (-0.01bpp au mieux). L'exécutable de SPIHT à disposition ne permet pas de fournir un débit supérieur à 8bpp et est donc difficilement comparable. Sur les deux séries de débit inférieur à 8bpp (avec des bandes noires) il est entre +0.05bpp et +0.1bpp et reste moins performant que JPEG-LS. Enfin JPEG2000 est à environ +0.3bpp.

6.2 Compression multi-coupes

Dans le but de mettre en évidence les techniques les plus à même d'être efficaces et pratiques pour la compression d'images médicales, quelques algorithmes tentant d'exploiter la corrélation volumique de l'information ont été étudiés.

Afin d'avoir une référence pour la décorrélation inter-coupe, les codeurs intra ont également été utilisés pour le codage de l'erreur résiduelle d'une prédiction DPCM inter-coupe simple : $\hat{I}(k) = I(k-1)$. Par la suite, ils seront appelés JPEG-LS-DIFF, JPEG2000-DIFF et SPIHT-DIFF.

Pour évaluer l'impact d'une décorrélation inter-coupe incluant une compensation de mouvement par blocs, nous avons utilisé une adaptation sans perte pour des vidéos sur plus de 8bits du codeur vidéo WAVIX [VGP02, BFG05].

Les ondelettes étant le plus souvent utilisées dans les travaux de compression d'images volumiques médicales, et afin d'évaluer les gains obtenus à l'aide d'une telle approche, deux codeurs disponibles sur le web ont été utilisés : JPEG2000+3D[1] (JPEG 2000 Part 2 muni d'une transformée en ondelettes 5/3 pour la décorrélation inter-coupes), et SPIHT3D, une extension volumique de SPIHT[2]. Ils restent basés sur les principes des algorithmes 2D dont ils sont l'extension, et sont donc plus facilement comparables avec les résultats de ces même algorithmes bidimensionnels.

Dans l'optique de permettre un accès aléatoire assez rapide, les codeurs ondelettes ont été appliqués sur des GOP de taille 16 resp. 32 pour une décomposition sur 5 niveaux en X,Y et 4 resp. 5 niveaux en Z. En règle générale un GOP de taille 32 n'a pas apporté de gain de compression supplémentaire. Sur les IRM (peu corrélées) et les scanners bruités, le nombre de bits par voxel (bpv) d'un GOP de taille 32 est quasiment égal à la moyenne des bpv des deux GOPs de taille 16 lui correspondant. Un léger gain a été noté sur les scanners propres (peu bruités/filtrés) inférieur à 0.05bpv (0.02bpv pour SPIHT3D et 0.01 pour JPEG2000+3D en moyenne). Afin de rendre les courbes légèrement plus lisibles, les résultats pour les GOPs de taille 32 ne seront donc pas présentés. Une sélection de courbes est disponible en Annexe B.

[1] Kakadu : http://www.kakadusoftware.com/Downloads.html
[2] http://www.cipr.rpi.edu/research/SPIHT/spiht3.html

6.2.1 Résultats

Pour commencer, les résultats de WAVIX ont montré que la compensation de mouvement classique, par *bloc matching*, ne semble pas adaptée : les meilleurs taux de compression ont été obtenus pour les tailles de blocs les plus importantes : 64×64. (Dans cette configuration, l'information compressée nécessaire à la représentation du mouvement occupe moins de 1/1000 du débit). Seuls les résultats obtenus à l'aide de ces blocs 64×64 seront présentés.

La version de WAVIX utilisée comporte une transformée en S (ondelettes de Haar entières) orientée selon le mouvement déterminé par bloc matching afin de décorréler l'information inter-coupe. On peut donc supposer que les discontinuités ainsi que le bruit (pouvant être corrélé sans mouvement et se retrouvant décorrélé après translation du bloc) introduits lors de la décorrélation inter-coupes sont défavorables à leur compression par ondelettes, WAVIX utilisant JPEG2000 pour compresser les images des coefficients de Haar. Il est également beaucoup plus lent que les autres algorithmes à cause du *bloc matching*, mais le programme n'a pas été optimisé. La transformée en S n'est également pas très efficace sur les images bruitées.

Sur les séquences 8bits, les meilleurs taux de compression sont obtenus par JPEG-LS-DIFF. Ceux de SPIHT-DIFF et JPEG2000-DIFF similaires en moyenne, sont environ de +0.2bpp. JPEG2000+3D et SPIHT3D ont des taux de compression oscillant autour de ceux de SPIHT-DIFF et JPEG2000-DIFF, et restent en moyenne équivalents. Les algorithmes intra-coupe sont souvent moins performants, et WAVIX se situe entre eux et les algorithmes volumiques.

Pour les séquences 12bits, les résultats obtenus sont très dépendants du bruit et de sa corrélation inter-coupe. Ainsi pour les scanners ayant un bruit non corrélé et pour les IRMs, la compression intra-coupe reste plus intéressante que les algorithmes tentant d'exploiter une corrélation inter-coupe. Un léger gain proche de 0.2bpp est tout de même notable entre les codeurs 2D et leur extension volumique, soit une réduction du débit d'environ 4%. Quant à la compression intra des coupes *-DIFF, elle est la moins performante et génère un débit supérieur à celui obtenu avec une compression intra uniquement utilisant les mêmes algorithmes.

Pour celles ayant un bruit plus faible ou plus corrélé entre les coupes (scanners filtrés et/ou construits de manière plus adaptée), les techniques utilisant une décorrélation inter-coupe deviennent (logiquement) plus efficaces. Bien entendu, les résultats peuvent varier selon la nature de la reconstruction et les post-traitements. SPIHT3D offre souvent les meilleurs résultats et permet de gagner jusqu'à 1.25bpp sur CALIC (pris comme référence pour la suite), soit approximativement 10% en moyenne. JPEG2000+3D est un peu moins performant (gain de 9.5%). Bien que ne semblant pas adapté, WAVIX offre tout de même un gain de codage intéressant sur les scanners comportant un bruit moins marqué, par rapport à un codage intra uniquement (allant jusqu'à -0.3bpp comparativement à CALIC). Il se positionne entre JPEG2000+3D et CALIC.

Sur ces mêmes séquences, les résultats de la compression DPCM inter-coupe sont assez variables et reflètent l'impact des algorithmes de génération et/ou de filtrage des volumes. Ainsi SPIHT-DIFF (qui fournit souvent les meilleurs taux de compression devant JPEG2000-DIFF lui même devant JPEG-LS-DIFF) est relativement proche de JPEG2000+3D, sauf sur quelques séquences (scanners en provenance de DMP) pour lesquelles ils vient se positionner en tête, devant SPIHT3D. On constate également que la technique prédictive de LOCO-I qui est adaptée pour la détection de contours d'une image naturelle l'est un peu moins pour la prédiction d'une image d'erreurs (résidus de la prédiction inter).

Les IRM3D qui sont directement construite de manière volumique possèdent également un bruit texturé corrélé en inter-coupe. Le gain de l'approche volumique est alors également intéressant (5% à 10% de gain entre SPIHT3D et CALIC).

6.2.2 Analyse

En compression avec pertes, que la technique soit prédictive et/ou par transformée, le bruit quand il est de faible amplitude a tendance à être filtré (et supprimé) par la quantification. En compression sans perte, par contre, le bruit d'acquisition ne peut pas être ignoré, et se doit d'être pris en considération, principalement lors de la phase de décorrélation.

Exemple — 6.2.1

A titre d'exemple, si on se place dans un cas particulier où deux images successive de la séquence volumique sont identiques au bruit d'acquisition près. On considère alors les deux images $X = I + B_X$ et $Y = I + B_Y$ résultant toutes deux de l'acquisition d'une image I perturbée par un bruit additif uniforme indépendant ($\text{cov}(B_X, B_Y) = 0$ et $\text{cov}(I, B_X) = \text{cov}(I, B_Y) = 0$) à valeurs dans $\{-k, \ldots, k\}$. Alors un simple prédicteur $\hat{Y}(x, y) = X(x, y)$, optimal pour ces images dans le cas non bruité ($B_X = 0$, $B_Y = 0$, $\text{var}(Y - \hat{Y}) = 0$), génèrera une image résiduelle $R = Y - \hat{Y} = B_X - B_Y$ telle que

$$\text{var}(R) = \text{var}((I + B_X) - (I + B_Y)) = \text{var}(B_X - B_Y) = \text{var}(B_X) + \text{var}(B_Y) - 2\,\text{cov}(B_X, B_Y),$$

et comme B_X et B_Y sont indépendants, $\text{var}(R) = \text{var}(B_X) + \text{var}(B_Y)$. Cependant l'entropie $H(R)$ sera inférieure à $H(B_X) + H(B_Y)$, la distribution uniforme étant le pire des cas pour l'entropie, et $B_X - B_Y$ se rapprochant d'une loi normale (théorème de la limite centrale). En fait, la fonction caractéristique de la loi de probabilité suivie par R correspond au produit de convolution entre la fonction caractéristique de B_X et celle de B_Y, ici identiques et uniformes. Elle est donc triangulaire, et

$$H(R) = -\sum_{x=-2k}^{2k} \frac{2k+1-|x|}{(2k+1)^2} \log_2\left(\frac{2k+1-|x|}{(2k+1)^2}\right) \underset{k\gg 1}{\approx} \log_2(2k+1) + \frac{1/2}{\log(2)}.$$

Le codage des trois images (I, B_X, B_Y) (si on disposait d'un filtre parfait permettant de retrouver I, ou si I était à disposition) aurait un coût supérieur au codage des deux images (X, R) quand

$$H(I) + H(B_X) + H(B_Y) > H(X) + H(R),$$

soit

$$H(X) - H(I) < 2\log_2(2k+1) - H(R) \approx \log_2(2k+1) - \frac{1/2}{\log(2)},$$

dans le cas du bruit blanc additif. Et $H(B_X - B_Y)$ étant supérieur à $H(B_X) = H(B_Y) = \log_2(2k+1)$, si ce filtre parfait existait, il serait plus intéressant de coder (X, B_Y) ou (Y, B_X).

Dans le cas où B_X et B_Y suivent tous deux une loi gaussienne de variance σ^2, $R = B_X - B_Y$ suit également une loi gaussienne de variance $2\sigma^2$. $H(B_X) = H(B_Y) = \frac{1}{2}\log_2(\sigma^2 2\pi e)$ et

$$H(R) = \frac{1}{2}\log_2(2\sigma^2 2\pi e) = \frac{1}{2}\log_2(\sigma^2 2\pi e) + \frac{1}{2}.$$

Ainsi il devient moins rentable de coder X et R que de coder séparément I, B_X et B_Y lorsque $H(X) + H(R) > H(I) + H(B_X) + H(B_Y)$, soit lorsque $H(X) - H(I) > \frac{1}{2}\log_2(\sigma^2\pi e)$.

Quoi qu'il en soit, si on dispose d'un filtre adapté f, permettant de supprimer en partie le bruit de X : $f(X) \approx I$, alors le prédicteur $\hat{Y}(x, y) = f(X)(x, y)$ devrait permettre de faire tendre $H(R)$ vers $H(B_Y)$, et ainsi s'approcher d'un codage optimal.

Cet exemple très simple ne correspond pas tout-à-fait aux conditions de compression des images volumiques : il est très rare d'avoir des coupes successives identiques et le bruit ne suit pas nécessairement une loi indépendante de l'image et indépendante entre des images successives. Cependant, l'amplitude du bruit et sa loi de probabilité peuvent nécessiter d'être pris en considération lors de la décorrélation inter-coupes.

Exemple — 6.2.2

Soient $X = I_X + B_X$ et $Y = I_Y + B_Y$, deux coupes d'une séquence (I_X et I_Y) et leur bruit d'acquisition (B_X et B_Y) indépendant ($\text{cov}(I_X, B_X) = \text{cov}(I_Y, B_Y) = 0$). Leur corrélation est :

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X)\,\text{var}(Y)}},$$

avec

$$\begin{aligned}\text{cov}(X, Y) &= \text{cov}(I_X + B_X, I_Y + B_Y) = \text{cov}(I_X, I_Y) + \text{cov}(B_X, I_Y) + \text{cov}(B_Y, I_X) + \text{cov}(B_X, B_Y)\\ &= \text{cov}(I_X, I_Y) + \text{cov}(B_X, B_Y),\end{aligned}$$

et

$$\begin{cases} \operatorname{var}(I_X + B_X) = \operatorname{var}(I_X) + \operatorname{var}(B_X) + 2\operatorname{cov}(I_X, B_X) = \operatorname{var}(I_X) + \operatorname{var}(B_X) \\ \operatorname{var}(I_Y + B_Y) = \operatorname{var}(I_Y) + \operatorname{var}(B_Y) + 2\operatorname{cov}(I_Y, B_Y) = \operatorname{var}(I_Y) + \operatorname{var}(B_Y) \end{cases},$$

d'où

$$\operatorname{corr}(X,Y) = \frac{\operatorname{cov}(I_X, I_Y) + \operatorname{cov}(B_X, B_Y)}{\sqrt{\operatorname{var}(I_X)\operatorname{var}(I_Y) + \operatorname{var}(B_X)\operatorname{var}(I_Y) + \operatorname{var}(B_Y)\operatorname{var}(I_X) + \operatorname{var}(B_X)\operatorname{var}(B_Y)}}.$$

Ainsi, plus les régions sont homogènes ($\operatorname{var}(I_X)$ et $\operatorname{var}(I_Y)$ faibles), plus la variance du bruit a une influence sur la corrélation, particulièrement si celui-ci est indépendant ($\operatorname{cov}(B_X, B_Y) = 0$).

Lorsque la corrélation est faible une prédiction inter-coupe n'est pas pertinente. Ainsi un simple prédicteur linéaire $\hat{Y} = X$ ne sera pas ou peu efficace dans des zones homogènes et risque de faire augmenter l'entropie, à moins que le bruit ne soit corrélé d'une coupe à l'autre ou soit de faible amplitude (variance).

Tout comme pour l'exemple 6.2.1, un filtrage de X, avant la prédiction, réduirait l'impact du bruit sur la corrélation.

Ce problème de non-corrélation des zones homogènes est soulevé par Van Assche, De Rycke, Philips et Lemahieu dans [vAdRPL99] qui font remarquer que la corrélation la plus importante entre des coupes successives se situe principalement au niveau des contours. Leur réponse est une adaptation volumique de JPEG-LS n'utilisant qu'une prédiction intra-coupe munie d'une modélisation de contexte intra et inter-coupe pour le codage entropique. Dans un esprit similaire, De Rycke et Philips [dRP99, PvAdRD01] ont également proposé d'utiliser une prédiction intra-coupe (JPEG-LS) sur la majorité des pixels de l'image sauf ceux pour lesquels le contexte de prédiction permet de détecter une variation trop importante (contours). Dans ce cas une compensation de mouvement inter-coupe est appliquée.

Il ne faut pas espérer gagner en compression en transmettant le bruit séparément et en décorrélant simplement l'information utile (image débruitée ou bits de poids fort), cependant avec une décorrélation utilisant des ondelettes entières, cette approche devrait permettre une meilleur progressivité, le bruit ayant une amplitude amplifiée par ces ondelettes non orthonormales.

La base d'images utilisée pour leurs travaux est la NLM-VHP (cf. section 1.4) et possède effectivement un bruit d'amplitude importante et non corrélé entre coupes successives. Ainsi, sur cette base, une décorrélation non supervisée n'entraine aucun gain de codage, mais plutôt des pertes. Même les techniques utilisant une décorrélation inter-coupes de type ondelettes ont des taux de compression inférieurs à un codage prédictif uniquement intra (CALIC et JPEG-LS) (cf.résultats Annexe B figure C.2).

Bien que la majorité des IRMs à disposition aient elles aussi un bruit très peu corrélé entre les coupes, les IRM3D, ont des coupes plus fine et plus corrélées (bruit également) grâce à leur reconstruction à l'aide d'un transformée de fourrier volumique. De même, sur la plupart des volumes tomographiques utilisés (DMP, base Osirix) le bruit est d'assez faible amplitude et suffisamment corrélé entre coupes successives pour qu'un simple prédicteur linéaire soit efficace.

Un certain nombre de situations qui pourraient être traitées différemment sont ainsi dégagées :
- bonne corrélation inter-coupe de l'image (coupes fines) :
  - corrélation inter-coupe du bruit faible (construction coupe à coupe, pas de filtrage) :
    La décorrélation ne doit être effectuée que dans des zones peu homogènes (fonction de l'amplitude du bruit), comme au niveau des contours ([vAdRPL99, PvAdRD01]). On peut aussi envisager de débruiter l'image, décorréler l'image débruitée et coder le bruit indépendamment afin de permettre une meilleur progressivité, ou encore utiliser une image débruitée pour effectuer une prédiction inter-coupe.
  - corrélation inter-coupe du bruit suffisante (construction volumique et/ou filtrage) :
    La décorrélation peut être effectuée sur la totalité des images.
- corrélation inter-coupe de l'image moyenne (coupes épaisses ou éloignées) :
  Un modèle de compensation de mouvement et/ou de distorsion d'objets peut être envisageable.
- corrélation inter-coupe de l'image faible ou insuffisante (coupes trop épaisses ou trop éloignées) :
  Un codage intra uniquement peut s'imposer. Les informations volumiques peuvent tout de même permettre de mieux modéliser le bruit et de le prendre en compte pour un codage entropique contextuel.

6.3 Régions d'intérêt

6.3.1 Extraction automatique

Afin de pouvoir utiliser les régions d'intérêt sur une base d'images importante, une technique rudimentaire d'extraction a été mise en place.

Dans un premier temps, l'image est binarisée à l'aide d'un seuillage d'histogramme. L'hypothèse prise est que l'image comprend un arrière plan de taille suffisante, et que cet arrière plan sera le plus sombre (puisque ne contenant pas d'information) et bruité (cette hypothèse se vérifie sur la majorité des clichés, sauf lorsqu'un recadrage serré a été effectué autour d'une région interne au patient). Ainsi si l'on considère l'histogramme comme un mélange de fonctions assimilables à des *lobes* (gaussiennes, fonction paraboliques, ...), le premier lobe (le plus proche de 0) doit correspondre à la couleur de l'arrière plan et son bruit d'acquisition et/ou de reconstruction. On cherche donc le point d'intersection entre ce lobe et le suivant. Ainsi tous les niveaux de gris compris entre zéro et cette intersection sont considérés comme ne faisant pas parti de la région d'intérêt. Un masque binaire est alors réalisé (1 pour les pixels supposés appartenir à la région d'intérêt, 0 sinon). Des opérations morpho-mathématiques lui sont appliquées afin de le « nettoyer » d'éviter la perte d'informations dans des zones internes à la région d'intérêt.

L'histogramme H est construit à l'aide des niveaux de gris de tous les voxels de l'image volumique dont on veut extraire la région d'intérêt. La longueur L de H est alors $L = 2^b$ (où b est le nombre de bits par niveau de gris). La valeur 0 de l'histogramme ($H(1)$), qui correspond à l'encadrement du disque de reconstruction pour les scanners ou des bandes noires sur certaines IRMs, est ignorée. Afin de rendre l'histogramme plus stable (suppression des hautes fréquences), un filtrage est appliqué sur celui-ci (cf. figure 6.3). Le filtre utilisé est un filtre convolutif triangulaire f dont la longueur F est ajustée en fonction de la répartition des valeurs de l'histogramme, et plus précisément de l'écart type de la distribution de probabilités correspondant à l'histogramme normalisé : $F = 2\lfloor\sqrt{\text{var}(H)}/50\rceil + 1$, où l'opérateur $\lfloor\cdot\rceil$ correspond à l'arrondi à l'entier le plus proche, et 50 a été fixé de manière empirique.

La variance est calculée classiquement :

$$\text{var}(H) = \frac{\sum_{k=2}^{L} H(k)(k - \text{cog}(H))^2}{\sum_{k=2}^{L} H(k)},$$

où $\text{cog}(H)$ est l'espérance mathématique de la distribution, qui correspond au centre de gravité de l'histogramme :

$$\text{cog}(H) = \frac{\sum_{k=2}^{L} k \cdot H(k)}{\sum_{k=2}^{L} H(k)}.$$

L'histogramme filtré H_f est alors obtenu à l'aide de produit de convolution de $H(k > 1)$ par f.

Afin de trouver l'intersection entre le premier lobe comprenant les valeurs de l'arrière plan et le suivant supposé intégrer des valeurs de la région d'intérêt, l'indice du premier maximum local x_{max} de H_f est conservé, tel que :

$$\forall x_0, x_1 \,/\, x_1 \leq x_{max} \text{ et } x_0 \leq x_1, \begin{cases} H_f(x_{max}) > H_f(x_{max}+1), \\ H_f(x_1) \leq H_f(x_{max}), \\ H_f(x_0) \leq H_f(x_1). \end{cases}$$

C'est le niveau de gris le plus représentatif des valeurs de l'arrière plan. Le minimum local suivant $x_{min} > x_{max}$ tel que :

$$\forall x_0, x_1 \,/\, x_{max} \leq x_1 \leq x_{min} \text{ et } x_{max} \leq x_0 \leq x_1, \begin{cases} H_f(x_{min}) < H_f(x_{min}+1), \\ H_f(x_1) \geq H_f(x_{min}), \\ H_f(x_0) \geq H_f(x_1). \end{cases}$$

est considéré comme point d'intersection entre les lobes (cf. figure 6.4).

Pour chaque coupe $C_k(i,j)$, un masque binaire $B_{0k}(i,j)$ est alors généré :

$$B_{0k}(i,j) = \begin{cases} 1 & \text{si } C_k(i,j) > x_{min}, \\ 0 & \text{sinon.} \end{cases}$$

Ce masque binaire contient souvent des résidus de bruit de l'arrière plan (on peut également trouver des lignes courbes d'épaisseur assez fines sur les coupes des scanners, qui correspondent à un tissu extérieur :

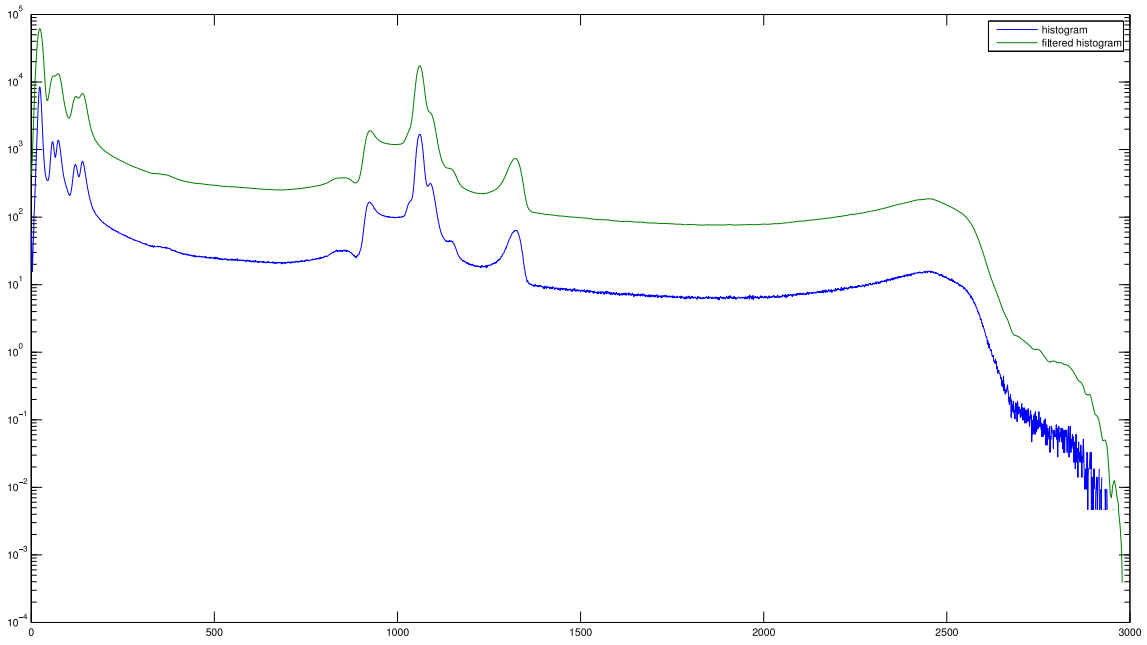


FIG. 6.3 – Histogramme H (bleu) et histogramme filtré H_f sans prise en compte de $H(1)$ (vert).

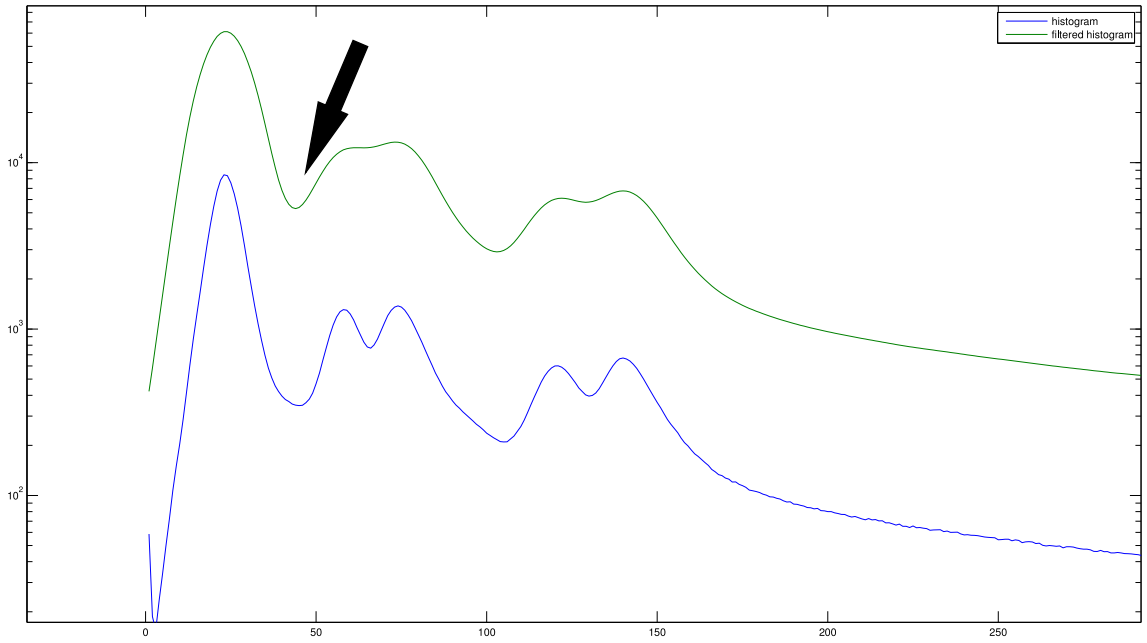


FIG. 6.4 – Histogramme H (bleu) et histogramme filtré H_f sans prise en compte de $H(1)$ (vert). Zoom sur le minimum local recherché.

vêtement du patient ou autre). Ces données ne sont pas nécessaires et une partie d'entre elles peuvent être supprimées à l'aide des morpho-mathématiques. On effectue donc une *ouverture* sur chaque coupe à l'aide d'un petit élément structurant s_o de taille 5×5 (disque de diamètre 5 s'apparentant à un losange pour ces dimensions) afin de supprimer le bruit et conserver les contours des régions d'intérêt :

$$B_{1k} = \text{open}(B_{0k}, s_o).$$

Enfin pour rendre le masque plus compacte, une *fermeture* utilisant un élément structurant s_c de taille 19×19 (disque de diamètre 19) est appliquée sur B_{1k} afin d'obtenir le masque B_{ROIk} utilisé par la suite pour définir la région d'intérêt :

$$B_{ROIk} = \text{close}(B_{1k}, s_c).$$

Les taille des filtres ont été fixées de manière empirique. A titre d'exemple d'application de ces opérateurs, on peut se référer aux figures 6.5 et 6.6, ainsi que la figure 6.7 pour constater la localisation de la région d'intérêt ainsi extraite.

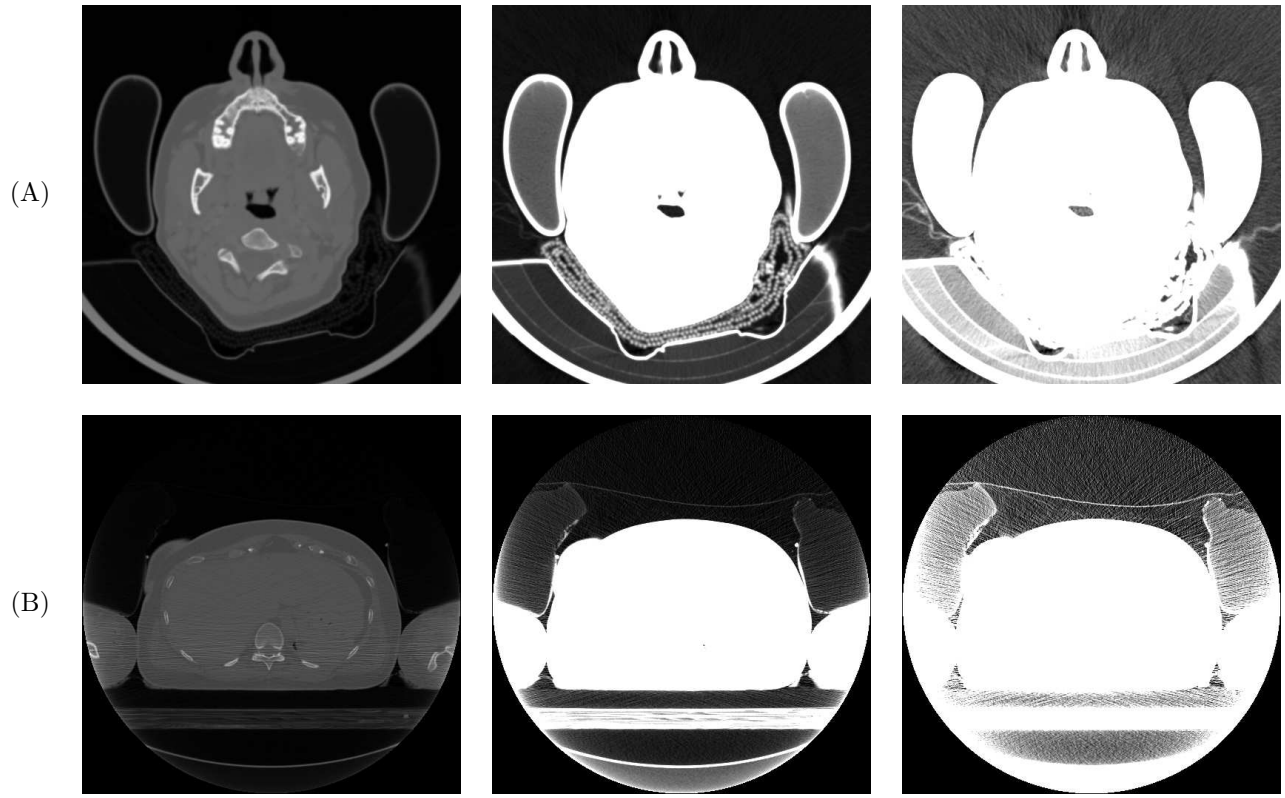


FIG. 6.5 – images utilisées pour les exemples d'opérations morpho-mathématiques (cf. figure. 6.6) ; ligne 1 : image (A) utilisée pour illustrer la binarisation à partir de l'histogramme, ligne 2 : image (B) plus bruitée que (A) ; colonne 1 : plage des valeurs entre 0 et val. max réajustée sur 256 niveaux de gris, colonne 2 : plage des valeurs entre 0 et val. max/8, colonne 3 : plage des valeurs entre 0 et val. max/32.

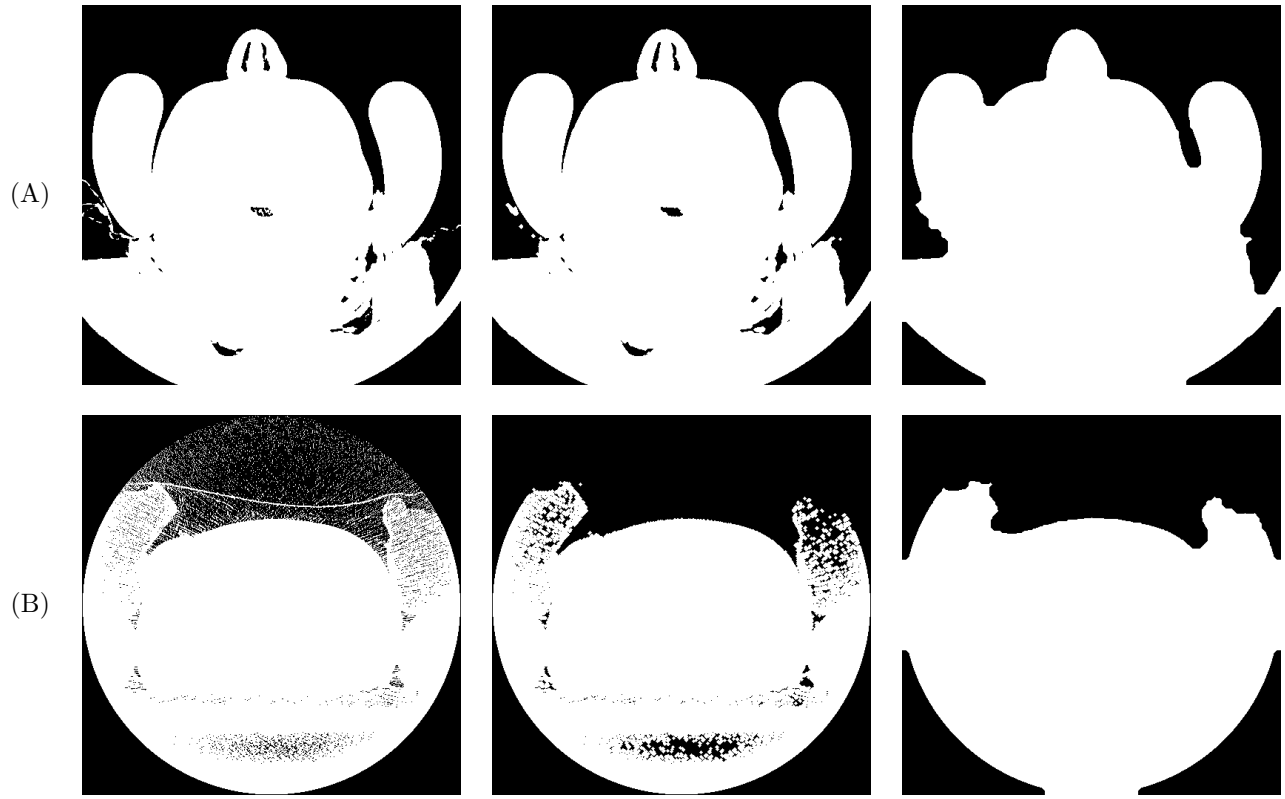


FIG. 6.6 – illustration des opérations morpho-mathématiques : ligne 1 : image (A), ligne 2 : image (B) plus bruitée que (A) (cf. figure 6.5) ; colonne 1 : image binarisée B_{0k} à l'aide de l'histogramme, colonne 2 : masque binaire B_{1k} après l'opération d'ouverture, colonne 3 : masque final B_{ROIk} après fermeture.

(A)

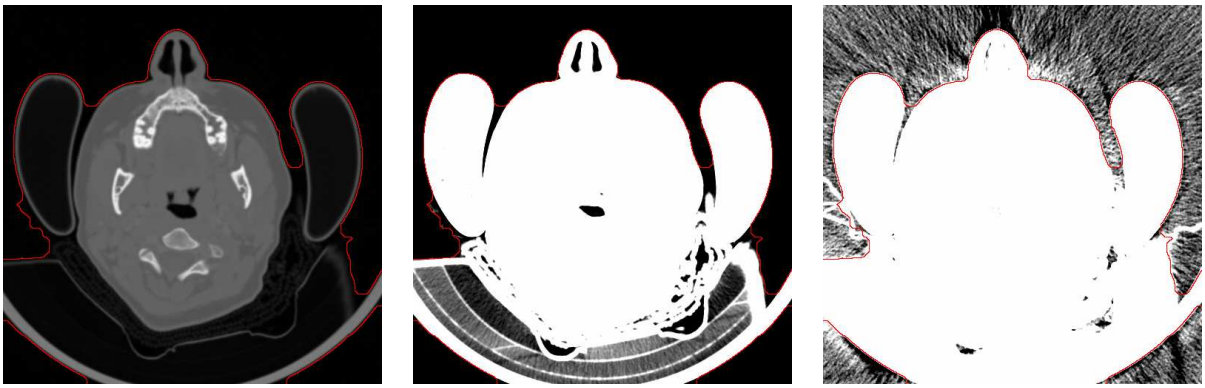


FIG. 6.7 – Illustration de la localisation de la région d'intérêt définie par le masque $B_{ROI\,k}$ de l'image (A) (cf. figure 6.6) en affichant différentes plages de valeurs. Cette zone est délimitée par la courbe rouge.

Bien entendu, la prise en compte des coupes successives (volumes) pour les opérations morpho-mathématiques garantirait davantage de fiabilité au procédé, mais les résultats coupe par coupe sont suffisamment convenables pour les expérimentions présentées par la suite.

Cette technique d'extraction de région d'intérêt reste simple mais suffisante pour pouvoir effectuer une analyse de l'impact de ces régions en compression à l'aide d'une base d'images importante. Comme on peut constater, les régions englobent des zones non significatives pour un diagnostique mais aucune information importante n'est perdue. Ainsi les résultats de compression de la zone d'intérêt qui seront présentés ne pourront être que meilleurs si une extraction plus fine est effectuée.

6.3.2 Méthodologie

Deux approches ont été étudiées, l'une utilisant une région d'intérêt différente pour chaque coupe, l'autre utilisant une région d'intérêt $B_{ROI\,(k,\ldots,k+N-1)}$ incluant l'ensemble des régions d'intérêt de chaque coupe dans un GOP de taille $N = 16$:

$$B_{ROI\,(k,\ldots,k+N-1)}(i,j) = \bigoplus_{m=k}^{k+N-1} B_{ROI\,m}(i,j),$$

où l'opérateur $\bigoplus$ correspond au OU en logique binaire.

Afin d'éviter de devoir mettre en place des variantes des codeurs précédemment utilisés (dont les codes sources ne sont pas tous disponibles) qui soient adaptés à la compression de régions d'intérêt, deux images sont générées : l'une comportant tous les pixels appartenant à la région (I_{ROI}) et l'autre incluant ceux hors de la région (I_{RONI}). Tous les pixels ignorés sont mis à 0 :

$$\mathrm{I}_{ROI}(i,j) = \begin{cases} \mathrm{I}(i,j) & \text{si } (i,j) \in ROI, \\ 0 & \text{sinon.} \end{cases}$$

$$\mathrm{I}_{RONI}(i,j) = \begin{cases} \mathrm{I}(i,j) & \text{si } (i,j) \notin ROI, \\ 0 & \text{sinon.} \end{cases}$$

Ainsi, encore une fois, les taux de compression présentés seront légèrement moins bons que si des codeurs prenant en compte la région d'intérêt lors de la décorrélation avaient été développés.

6.3.3 Résultats et analyse

La compression des masques binaires à l'aide de CALIC ou JPEG-LS a nécessité moins de 0.05 bits par pixel sur les quelques clichés analysés. Cette quantité d'information étant suffisamment négligeable par rapport aux taux de compression des images, un codage plus performant de la localisation de la région d'intérêt n'a pas été considéré et il sera ignoré par la suite. De plus, la compression sans perte totale consistant à compresser la ROI puis la RONI a l'aide de cette technique n'est évidemment pas idéale, les pertes de compression par rapport à une approche monolithique pouvant varier entre +0.1 et +0.5 bits par pixels selon les algorithmes et les représentations des régions d'intérêt. Les images de la RONI deviennent plus difficiles à compresser, principalement lorsqu'un masque est utilisé pour chaque coupe.

En effet, les algorithmes peuvent être perturbés par la RONI qui est souvent segmentée en plusieurs zones non connexes : la transition entre une zone contenant des donnée et une région ignorée (les valeurs mises à 0) peut générer quelques coefficients/erreurs de prédiction d'amplitudes plus élevées. De même le codage des pixels des de ces régions ignorées a un coût.

L'objectif est d'évaluer les gains potentiels sur la taille des fichiers compressés en supprimant une information totalement inutile au diagnostique. Ainsi, avec l'utilisation de codeurs ne prenant pas en compte le masque binaire, celui-ci n'a pas besoin d'être transmis pour la décompression. Quelques exemples de résultats obtenus pour la région d'intérêt par coupes sont fournis en Annexe C.

Sur les images avec une bonne résolution transversale (z), la compression par ondelettes volumiques (SPIHT3D et JPEG2000+3D) de la région d'intérêt dans des GOP de taille 16 fournit des résultats assez similaires, que la ROI soit unique au GOP ou locale à chaque image. Une légère perte de performances est notable pour la ROI unique, lorsque la résolution en (z) est plus faible et que les ROI locales sont trop changeantes (peu de corrélation inter-coupes) : plus d'information inutile doit être compressée.

Comme pour la compression sans perte, les algorithmes volumiques sont plus performants lorsqu'il existe une certaine corrélation entre les coupes. JPEG-LS et CALIC en mode intra restent également plus performants lorsqu'un bruit trop important et non corrélé entre les coupes est présent (cf. section 6.2)

Le gain de compression apporté par l'utilisation de la ROI varie en fonction la taille de celle-ci. Sur des images bruitées, il est plus ou moins proportionnel au pourcentage de recouvrement de la RONI. Cette approche peut ainsi faire économiser de 20 à 40% d'espace de stockage par rapport à une compression sans perte classique.

6.4 Compression presque sans perte

Un autre travail d'investigation a été mené sur la compression presque sans perte de ces images médicales (cf. section 3.2.3). L'étude a porté sur l'évaluation de plusieurs techniques en mode intra-coupe, la majorité des travaux existant étant focalisés sur des images 2D.

Le PAE des images générées par SPIHT et JPEG2000 pour un taux de compression donné, ainsi que le standard JPEG-LS intégrant un mode presque sans perte (cf. 4.1.3), ont été utilisés comme références. Cependant, l'idée étant de proposer un représentation progressive (*lossy to lossless* ou *lossy to near-lossless*) de façon à pouvoir fournir un résumé de basse qualité (ou résolution) rapidement, les autres techniques comparées sont des approches multi-résolution. Ainsi un codeur tel que SPIHT peut être utilisé sur ces représentations afin de fournir une progressivité en qualité, et si une progressivité en résolution est nécessaire, un algorithme intra-bande tel que EBCOT (JPEG 2000) peut l'être également.

6.4.1 PQW

Trois approches différentes ont été comparées. La première, classique, consiste à quantifier uniformément les valeurs des pixels $v(x, y)$ de l'image en respectant le PAE δ. L'image des indices de quantification $v_q = Q_\delta(v)$ sont ensuite compressés sans perte (ici à l'aide de JPEG2000 et SPIHT), avec

$$Q_\delta(v) = \text{sign}(v) \left\lfloor \frac{|v| + \delta}{2\delta + 1} \right\rfloor . \tag{6.1}$$

La reconstruction se fait par déquantification $\tilde{v} = \tilde{Q}_\delta(v) = \bar{Q}_\delta(Q_\delta(v))$, avec

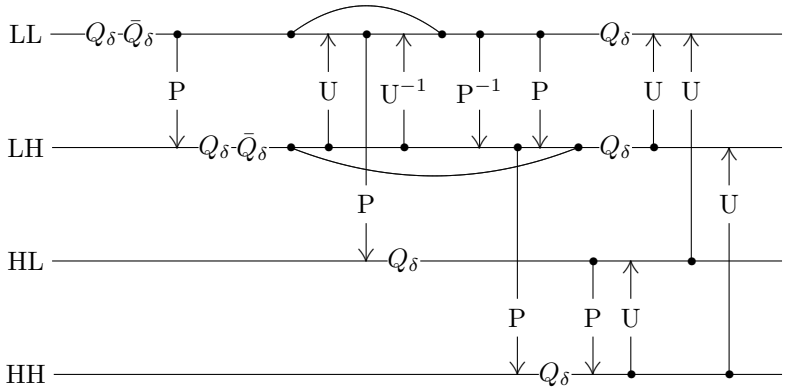
$$\bar{Q}_\delta(v_q) = (2\delta + 1)v_q. \tag{6.2}$$

Cette technique est parfois référencée sous le nom de PQW (*Pre-Quantized Wavelet*). Les résultats seront donc identifiés par JPEG2000-PQW et SPITH-PQW.

6.4.2 OQW

La seconde OQW (*Online Quantized Wavelet*), introduite par Charith Abhayaratne [Abh03], cherche à effectuer la quantification directement dans l'espace transformé en ondelettes. En se plaçant dans un schéma de lifting, une telle quantification n'est pas aisée à cause des relations existantes entre les coefficient des différentes échelles. Wu et Bao [WB97] utilisent une technique de programmation dynamique (assez coûteuse en temps CPU) afin de quantifier directement dans le domaine transformé et de s'approcher au mieux de l'erreur maximale souhaitée, puis ils utilisent CALIC pour compresser le résidu sur l'image nécessaire à atteindre ce PAE.

La méthode proposée par Abhayaratne est beaucoup plus simple. Elle utilise un schéma de lifting embarquant une quantification au premier niveau de décomposition. Ce schéma est le suivant :



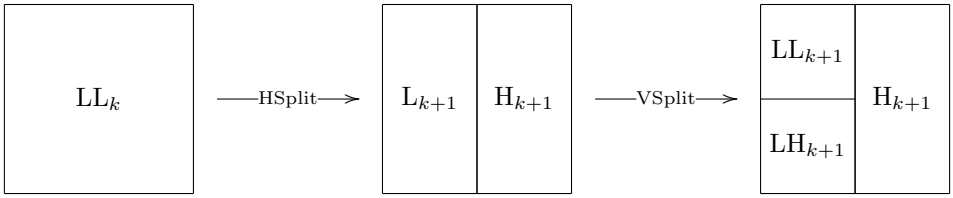
(6.3)

où (LL) correspond aux pixels de l'image $I(2x, 2y)$, (LH) aux pixels $I(2x, 2y+1)$, HL aux pixels $I(2x+1, 2y)$ et HH aux pixels $I(2x+1, 2y+1)$, et permettent d'effectuer une décomposition dyadique (cf. schéma 4.4). P, P^{-1}, U et U^{-1} sont les opérateurs classiques du lifting : prédiction, inversion de la prédiction, mise à jour et inversion de la mise à jour.

Ainsi l'image basse résolution (LL) du premier niveau d'une décomposition dyadique est d'abord quantifiée (Q_δ et $\bar{Q}_\delta$ correspondant aux opérateurs définis dans les équations 6.1 et 6.2). Elle est ensuite utilisée pour prédire les bandes hautes fréquences (LH, HL et HH). Enfin l'erreur de prédiction est également quantifiée et utilisée pour supprimer les hautes fréquences résiduelles dans les bandes LL, LH et HL. Ainsi, tous les pixels de l'image reconstruite ont une erreur respectant le PAE. Les niveaux suivants de la décomposition, appliqués sur la bande basses fréquences quantifiée, suivent alors un schéma de lifting classique n'intégrant aucune quantification ou perte supplémentaire.

6.4.3 QHI

La troisième technique utilisée, QHI (*Quantized Hierarchical Interpolation*) reprend le principe de l'algorithme HINT, mais utilise une décomposition dyadique (ligne,colonne), plutôt qu'une décomposition quinconce(ligne+colonne,diagonale+antidiagonale). L'image est tout d'abord sous-échantillonnée selon le schéma de décomposition suivant :



(6.4)

où HSplit est un sous-échantillonnage selon les lignes, et VSplit selon les colonnes. Cette décomposition est effectuée jusqu'à obtenir LL_0 correspondant à un unique pixel. L'image LL_0 est quantifiée, et sa valeur après déquantification est utilisée pour prédire LH_0. L'erreur de prédiction de LH_0 est quantifiée, et l'image *basse résolution* $\tilde{L}_0$ est reconstruite à partir de ces valeurs. $\tilde{L}_0$ est également utilisée pour prédire H_0 dont l'erreur de prédiction est également quantifiée. Une image $\tilde{LL}_1$ est ainsi obtenue et le processus est réitéré jusqu'à obtention d'une image $\tilde{I}$ respectant le PAE.

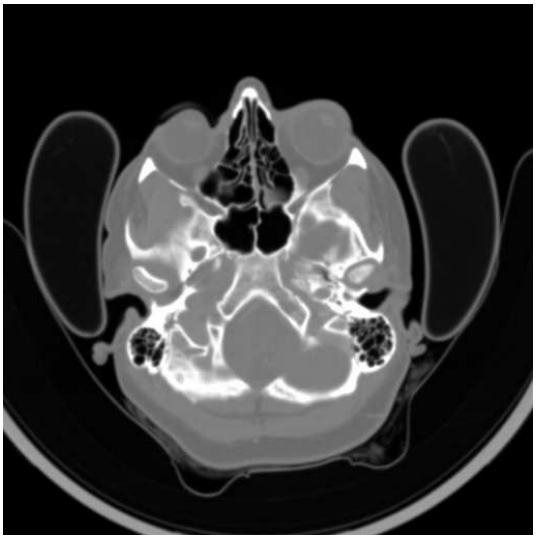
Avant de compresser cette décomposition, les coefficients d'erreur de prédiction quantifiés de H_k subissent une étape de lifting, permettant de retrouver une pyramide dyadique, et d'améliorer la décorrélation. Les coefficients peuvent ainsi être compressés à l'aide d'un algorithme pour coefficients d'ondelettes.

Une autre technique légèrement plus efficace (s'appuyant sur une décomposition IHINT : *Interleaved HINT* [AAB97]) consisterait à utiliser VSplit pour décomposer la bande H (non quantifiée) en HL et HH, utiliser $\tilde{Q}$(HL) pour prédire HH, pour enfin obtenir Q(HL) et Q(HH).

6.4.4 Méthodologie

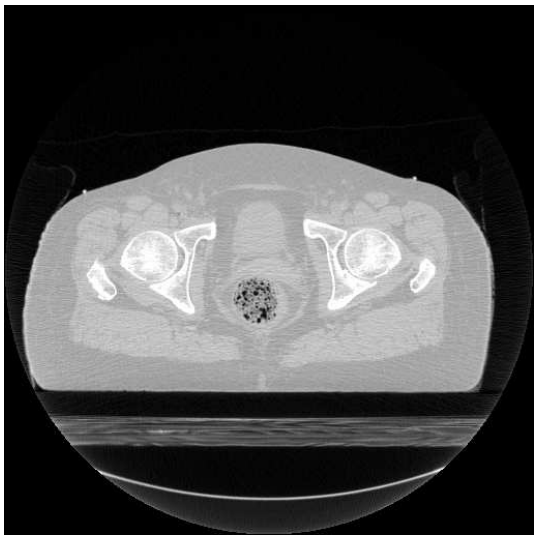
Une sélection d'images a permis d'évaluer les différentes techniques de compression presque sans perte (cf. figure 6.8). Cette sélection comprend des tomographies peu bruitées (DMP), avec un bruit plus important mais diffus (MeDEISA) et fortement bruitées (NLM-VHP), des IRMs et des IRMs-3D récentes (DMP).

L'algorithme OQW a été appliqué a l'aide d'un lifting 5/3. La prédiction utilisée pour QHI est identique à celle du lifting 5/3. Celui-ci est également utilisé pour la décorrélation des bandes d'erreurs H_k.

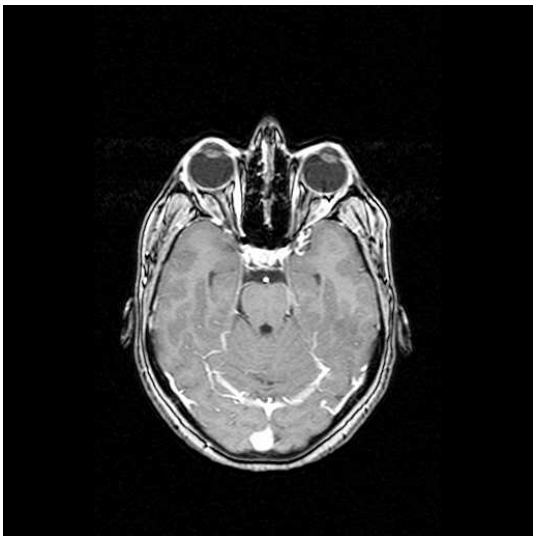


(A)

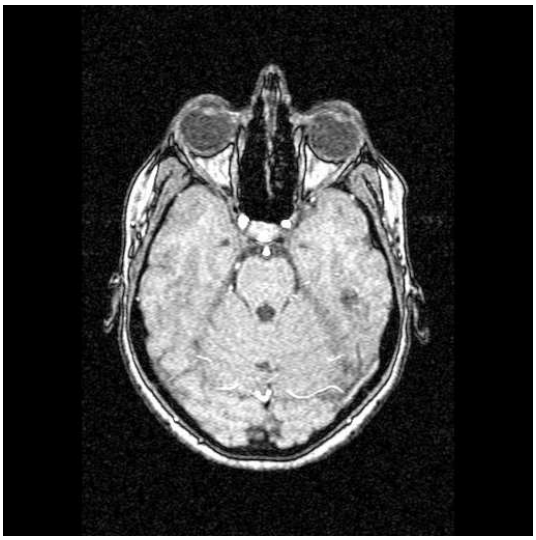
(B)



(C)



(D)



(E)

FIG. 6.8 – Exemples d'images de la sélection : (A) CT avec un bruit faible, (B) CT avec un bruit de reconstruction important mais diffus (filtrage), (C) CT avec un bruit très marqué, (D) IRM et (E) IRM3D. Le contraste des images a été ajusté pour leur visualisation en 256 nivaux de gris.

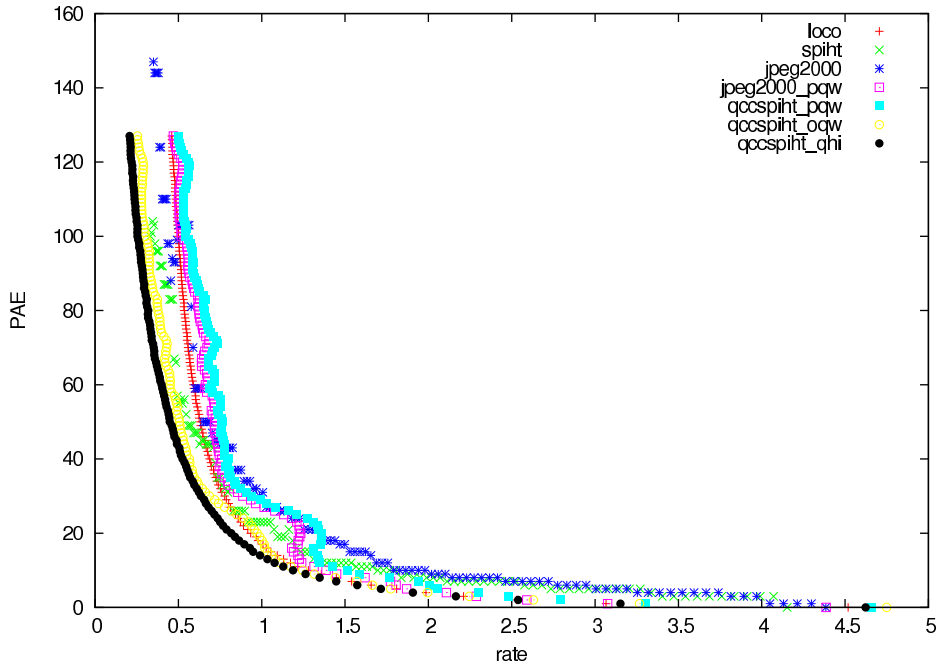


FIG. 6.9 – Résultats de la compression presque sans perte de l'image (A) de la figure 6.8

Un codeur SPIHT a été utilisé pour la compression des coefficients des trois différentes approches. Ce codeur provient d'une adaptation de l'implémentation fournie en module complémentaire de la bibliothèque de développement QccPack[3]. La version sans codage arithmétique a été préférée, l'autre ne fournissant pas des résultats corrects. Les résultats seront présentés sous les noms de SPIHT-PQW, SPIHT-OQW et SPIHT-QHI. Son implémentation diffère légèrement de celle fournie par Said et Pearlman disponible en version exécutable uniquement. Les taux de compression à haut débit (sans perte ou presque) son légèrement moins bons, mais quasiment identiques à bas débit.

6.4.5 Résultats

Les courbes débit/PAE seront utilisées pour comparer les résultats (se référer aux figures 6.9 à 6.13 pour avoir quelques illustrations). Bien que JPEG2000 semble plus performant que SPIHT pour la compression des images quantifiées (PQW), SPIHT ayant été utilisé avec les trois approches multi-résolutions, les comparaisons seront effectuées avec l'aide de ce dernier. A titre purement comparatif, les figures présentent également les courbes de la distorsion (PAE) produites lors de la compression des images à l'aide de JPEG 2000 et de SPIHT pour un débit fixé (l'optimisation effectuée par ces algorithmes tente de maximiser le PSNR).

Entre QHI, PQW et OQW, il ressort que SPIHT-QHI fournit presque toujours les meilleurs résultats. SPIHT-OQW offre des taux de compression très proches et SPIHT-PQW reste également convenable pour un δ faible, mais diverge assez rapidement ($\delta > 16$). PQW fournit des résultats peu stables, la quantification uniforme n'étant pas très adaptée pour l'espace de représentation image (espace des niveaux de gris) et pour la transformée en ondelettes qui suit. La même observation peut également être faite pour OQW qui quantifie tout de même 1/4 de ses valeurs sur une image basse résolution.

Lorsque les tomographies sont très bruitées, JPEG-LS offre les meilleurs performances débit/PAE, mais pour les autres types d'images il se fait rapidement devancer par SPIHT-QHI et SPIHT-OQW ($\delta \approx 16$). Bien que SPIHT et JPEG2000 ne soient pas conçus pour la norme L_∞, ils devancent parfois JPEG-LS($\delta > 20$).

Visuellement, lorsque la dégradation est assez forte (cf. figures 6.14 et 6.15), JPEG-LS, qui favorise l'encodage RLE, génère des trainées horizontales uniformes d'un pixel d'épaisseur. PQW réduit le nombre de niveaux de gris d'affichage et apporte donc un effet de tramage (transitions sèches, contenu moins riche). Cet effet apparait également, mais plus atténué, avec OQW qui restreint le nombre des niveaux de gris en quantifiant la bande basse fréquences. Enfin, QHI a tendance à produire des images comportant une palette de niveaux de gris (histogramme) beaucoup plus importante et mieux répartie. Cependant le fait

[3] http://qccpack.sourceforge.net/

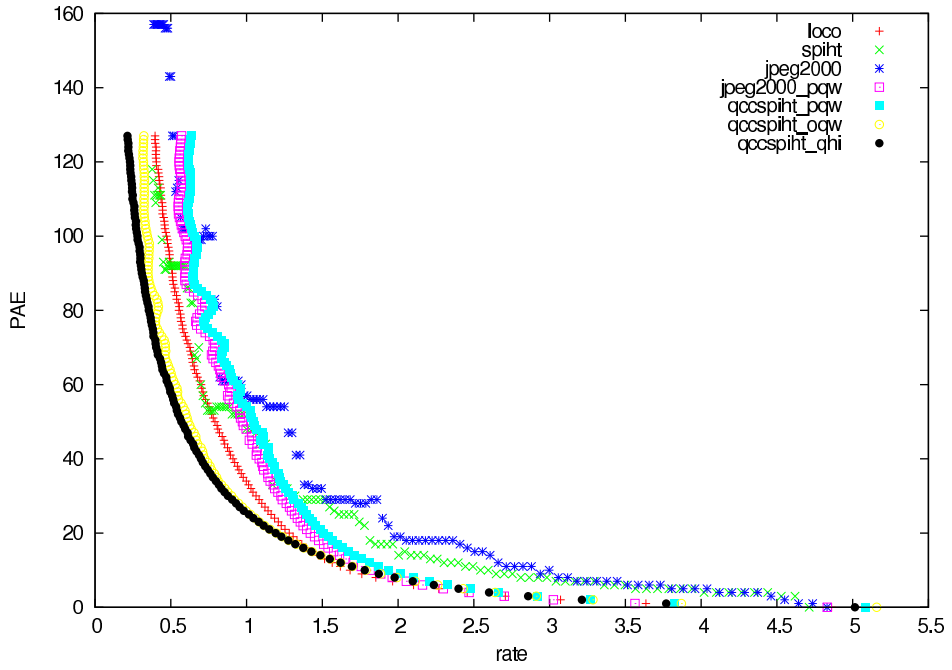


FIG. 6.10 – Résultats de la compression presque sans perte de l'image (B) de la figure 6.8

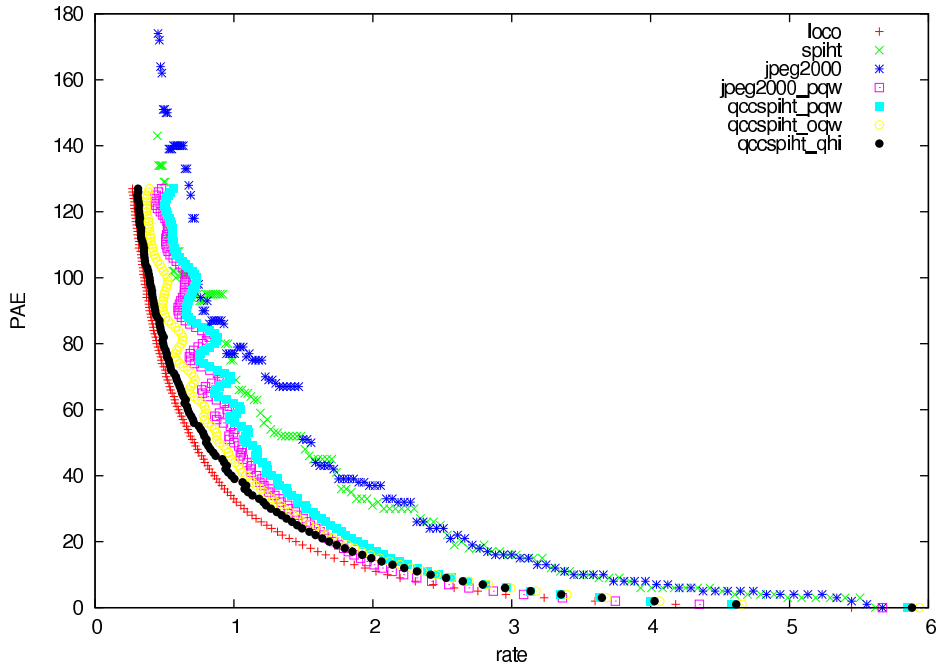


FIG. 6.11 – Résultats de la compression presque sans perte de l'image (C) de la figure 6.8

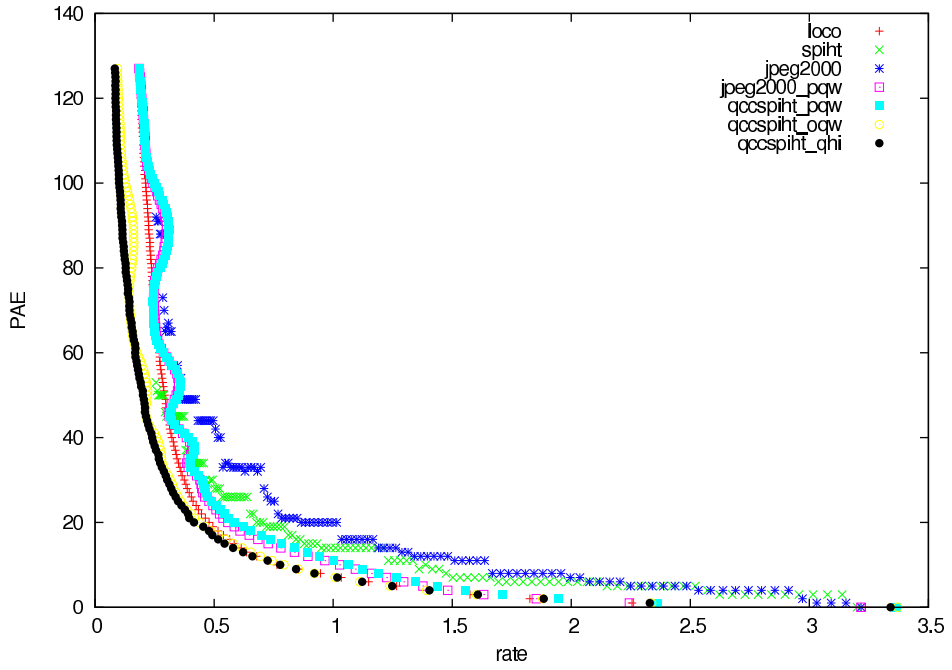


FIG. 6.12 – Résultats de la compression presque sans perte de l'image (D) de la figure 6.8

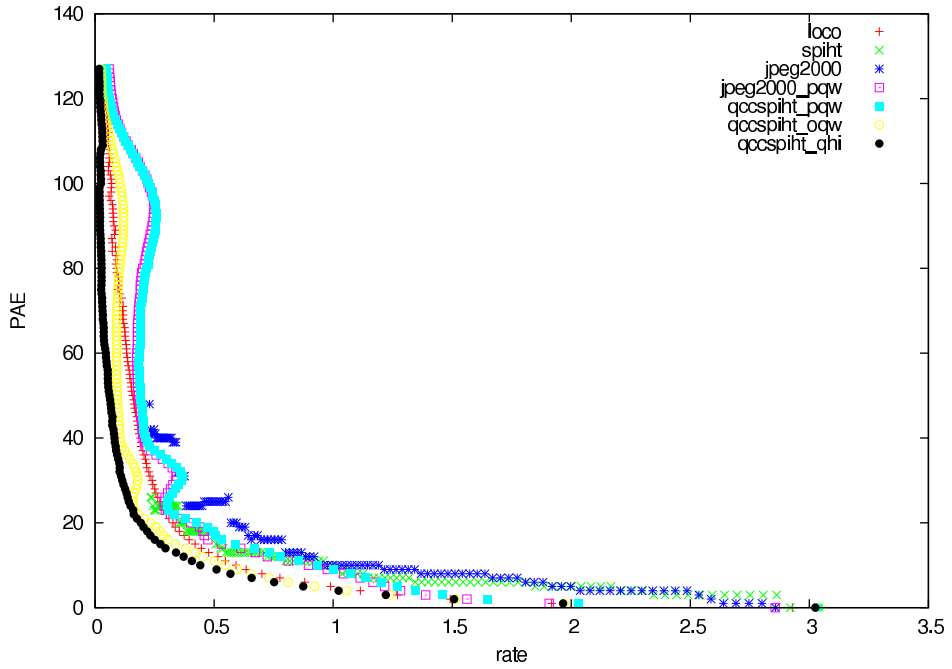


FIG. 6.13 – Résultats de la compression presque sans perte de l'image (E) de la figure 6.8

de sous-échantillonner sans filtrage préalable (et donc de ne pas respecter le théorème de l'échantillonnage) peut produire des artefacts pouvant être gênants. Des pics hautes fréquences (Dirac) peuvent apparaître par endroits.

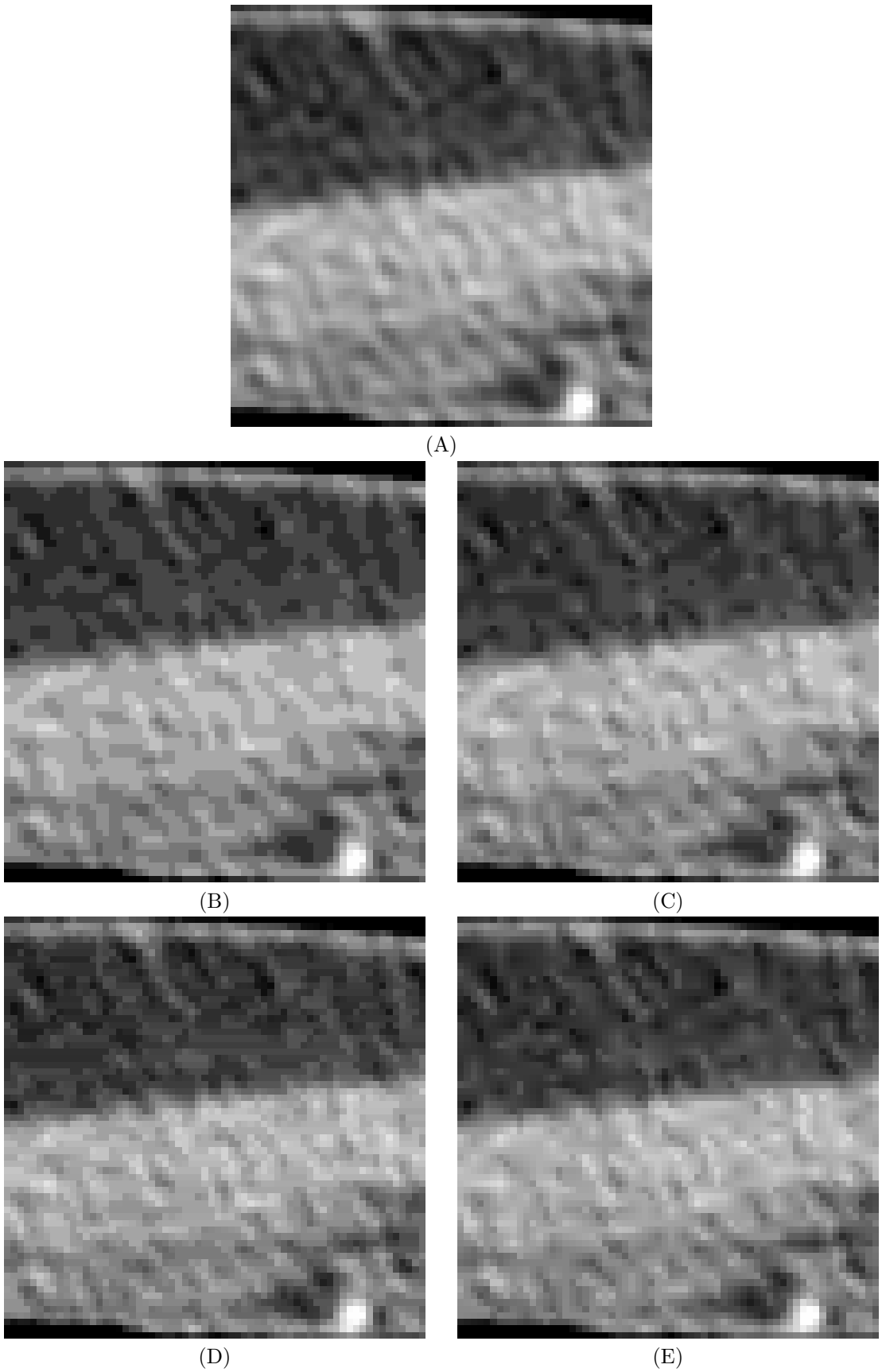
L'erreur maximale pouvant satisfaire les médecins est difficile à évaluer, et peut dépendre de la modalité, de l'organe ou de la pathologie. Cependant aux vues de l'amplitude du bruit, on peut estimer qu'une erreur de $\delta = 16$ serait tolérable. Les algorithmes étudiés sont quasiment équivalents pour $0 \leq \delta \leq 16$, cette tolérance d'erreur permettrait de gagner de 69% d'espace de stockage pour des scanner bruités à 87% pour des IRM3D, en comparaison à une compression sans perte. Les résultats présentés dans la figure 6.16 ont été obtenus avec JPEG-LS et ne sont que des estimations. Ils correspondent à la moyenne sur 3 images bien distinctes de chacune des séquences (A),(B),(C),(D) et (E) de la figure 6.8. Elles sont respectivement appelées SCAN, SCAN+, SCAN++, IRM et IRM3D. Ce tableau illustre le gain d'espace pouvant être espéré grâce à une compression presque sans perte, pour différents seuils d'erreurs δ.

Conclusion

Il ressort de ces expérimentations que les méthodes de construction et les filtres appliqués sur les images peuvent avoir un impact important sur les taux de compression. Ainsi, sur les images volumiques, si les coupes sont reconstruites une à une, ou de façon tridimensionnelle (IRM3D, ou approche volumique pour la reconstruction de tomographies), la corrélation inter-coupe varie énormément en particulier au niveau du bruit et des artefacts de reconstruction. Ceux-ci sont difficilement compressibles et ne doivent pas être négligés lors de la phase de décorrélation d'un schéma sans perte.

En compression volumique, pour la majorité des images qui ont un bruit intercoupe corrélé, une simple prédiction DPCM suivie d'un codage intra offre un gain important. Par contre pour les images dont le bruit n'est pas corrélé (majorité des IRMs, et quelques SCANs), la prédiction DPCM renforce le bruit (augmentation de la variance) et on observe des pertes de performances en comparaison à une compression intra-coupe uniquement. Sur ces images, les performances des codeurs volumiques utilisant des ondelettes sont légèrement meilleures que celles de l'algorithme 2D dont ils sont l'extension. Cependant leurs résultats sur ces images bruitées restent plus faibles que ceux des algorithmes prédictifs intra-coupes (CALIC, JPEG-LS). Des techniques de décorrélation plus robustes au bruit peuvent donc être envisagées et mises en place.

Malgré tout un gain d'espace de stockage allant de 20% à 40% peut être obtenu en supprimant l'information hors régions d'intérêt (voir plus avec des techniques de sélection de ROI plus avancées que celle présentée). En tolérant une faible erreur sur chacun des voxels (± 1 à ± 16), on peut également réduire la taille des fichiers de 20% à plus de 70%.



(A)

(B)

(C)

(D)

(E)

FIG. 6.14 – Illustration des artefacts générés avec les différentes méthodes avec $\delta = 16$: (A) image originale, (B) PQW, (C) OQW, (D) JPEG-LS, (E) QHI. Les patchs d'exemples sont extraits de l'image (B) de la figure 6.8 et de ses versions détériorées. Les paramètres de contraste et la luminosité sont les mêmes pour tous les patch et ont été ajustés de manière à mettre en évidence les distorsions

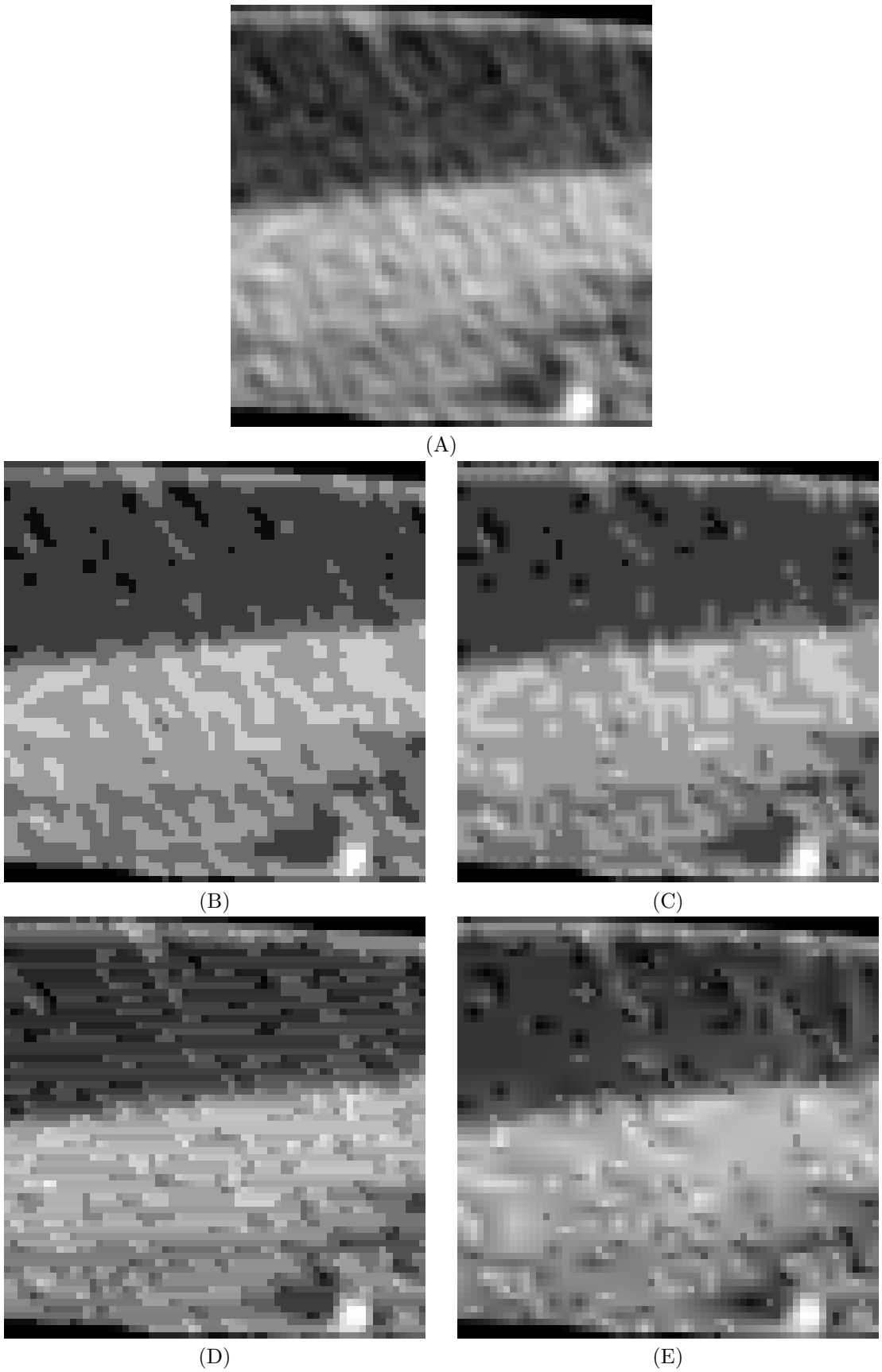


FIG. 6.15 – Illustration des artefacts générés avec les différentes méthodes avec $\delta = 32$: (A) image originale, (B) PQW, (C) OQW, (D) JPEG-LS, (E) QHI. Les patchs d'exemples sont extraits de l'image (B) de la figure 6.8 et de ses versions détériorées. Les paramètres de contraste et la luminosité sont les mêmes pour tous les patch et ont été ajustés de manière à mettre en évidence les distorsions

δ	SCAN++	SCAN+	SCAN	IRM	IRM3D
1	23	26	34	30	35
2	34	38	46	44	49
3	41	45	53	51	57
4	46	50	58	57	62
5	49	54	62	61	66
6	53	58	65	65	70
7	55	60	68	68	73
8	58	63	70	71	76
9	60	64	72	74	78
10	61	66	74	76	80
11	63	67	75	78	81
12	65	68	76	79	83
13	66	69	77	81	84
14	67	70	78	81	85
15	68	71	79	82	86
16	69	72	79	83	87

FIG. 6.16 – Pourcentages de la taille des fichiers compressés sans perte pouvant être économisés grâce à une compression presque sans perte. SCAN symbolise les scanners peu bruités, SCAN+ ceux avec un bruit plus important mais diffus et SCAN++ ceux avec un bruit très marqué. Ces résultats ont été obtenus avec JPEG-LS.

Chapitre 7

Conclusion

Ce document résulte d'un travail d'investigation sur la compression sans perte d'images médicales volumiques effectué durant une première année de doctorat. Nous y avons synthétisé les besoins dans ce milieu et présenté les deux modalités d'images qui semblent le plus nécessiter d'une compression efficace : les scanners et les IRMs. Nous avons dressé un état de l'art en compression sans perte d'images bidimensionnelles et volumiques, ces méthodes pouvant être destinées ou non à l'imagerie médicale. Cet état de l'art à donné lieu à une étude comparative des résultats obtenus pour différentes techniques de compression intra-coupe et après décorrélation volumique. Il en ressort que les taux de compression varient énormément en fonction des algorithmes de construction des images volumiques et/ou des post traitements qui leurs sont appliqués. Ainsi, le bruit et les artefacts de construction qui sont parfois diminués sur certains volumes voir corrélés entre coupes successive seront plus aisément compressés volumiquement. Sur d'autre images ce bruit peut être important, indépendant d'une coupe à l'autre et ainsi rendre les mêmes algorithmes moins efficaces qu'une compression intra-coupe uniquement. La résolution d'acquisition joue également un rôle important sur la corrélation inter-coupe des données ayant un intérêt diagnostique.

Le sans perte étant fortement pénalisé par le bruit qui est relativement présent dans ces types d'images, les travaux semblent s'orienter vers une compression avec pertes, ce qui n'est pas toujours en concordance avec l'éthique des médecins. Nous avons donc également étudié l'impact des pertes contrôlées sur la compression, que ce soit avec l'aide de régions d'intérêt, ou d'un seuil d'erreur maximal sur chacun des pixels. Ces approches permettent d'obtenir un gain d'espace de stockage intéressant tout en garantissant une certaine qualité aux médecins.

Appendices

Annexe A

Compression Intra

Cette annexe présente les courbes des résultats de la compression coupe par coupe (cf. section 6.1) des cinq volumes présentés dans la section 6.4. Ces cinq volumes ont été sélectionnés parmi un ensemble de plus de 100 volumes en provenance de différentes bases d'images (cf. section 1.4) sur lesquels ont portés nos investigations.

En abscisses les numéros de coupes et en ordonnées le nombre de bits par pixels utilisés après compression.

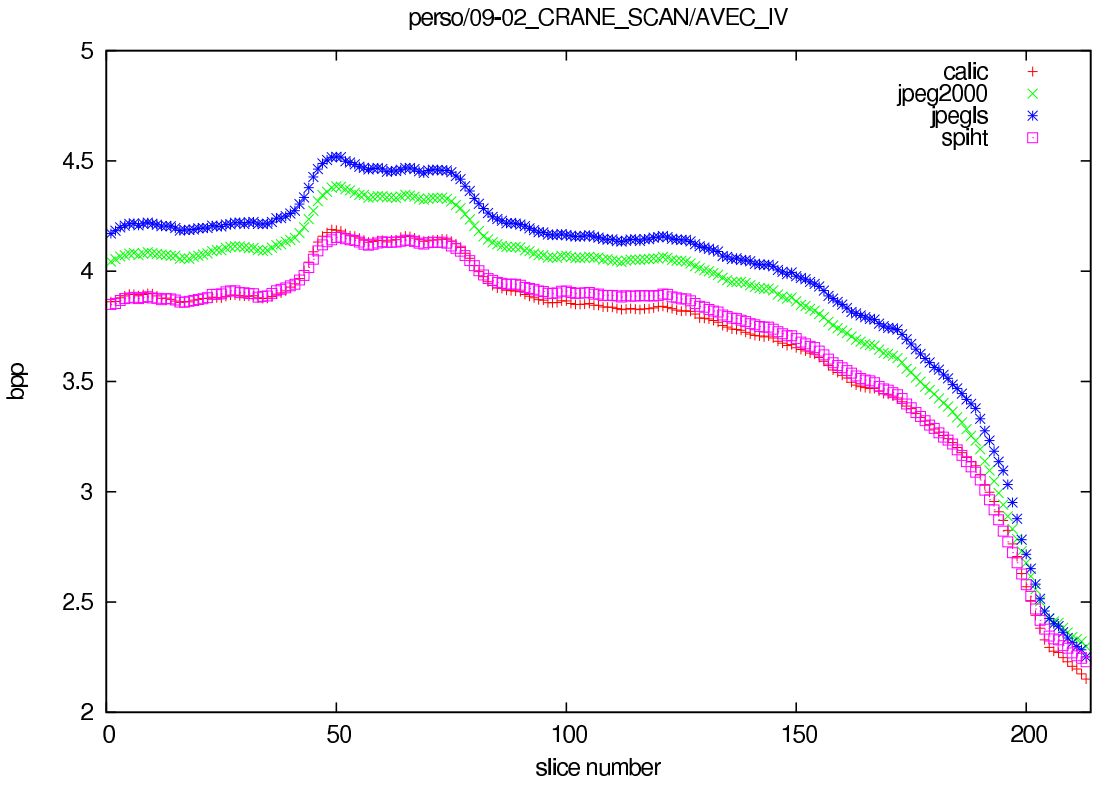


FIG. A.1 – volume (A), CT avec un bruit faible

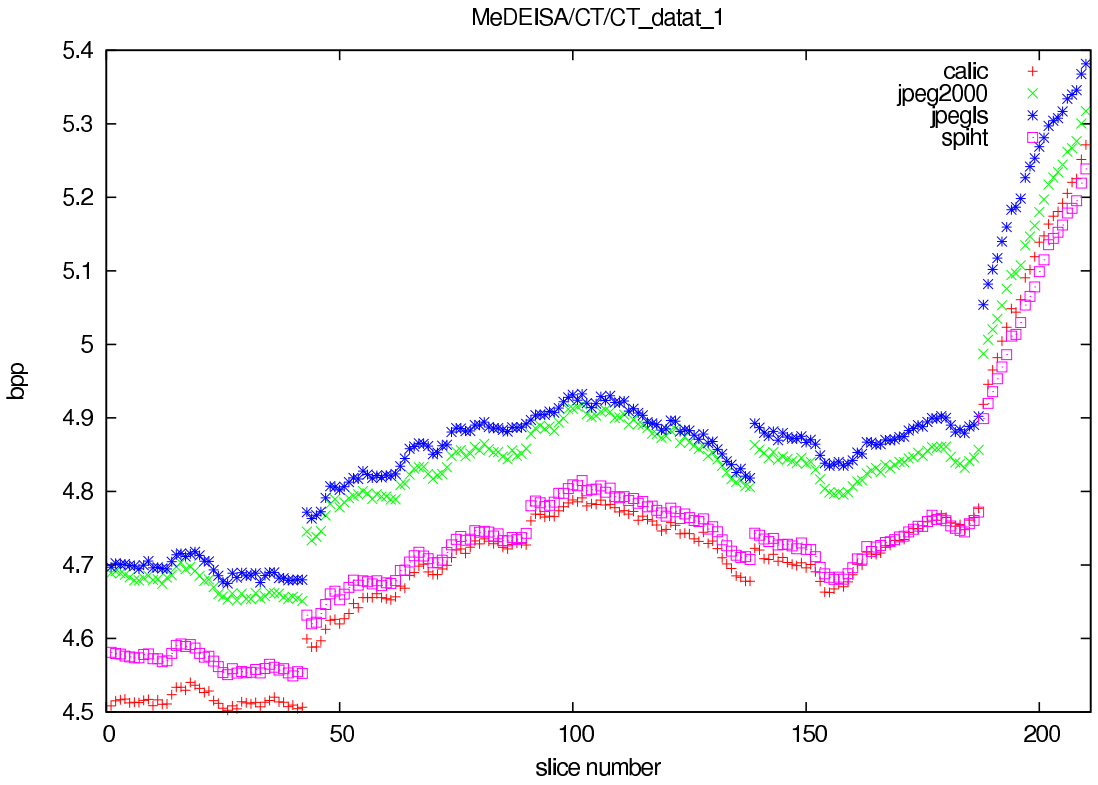


FIG. A.2 – volume (B), CT avec un bruit de reconstruction important mais diffus (filtrage)

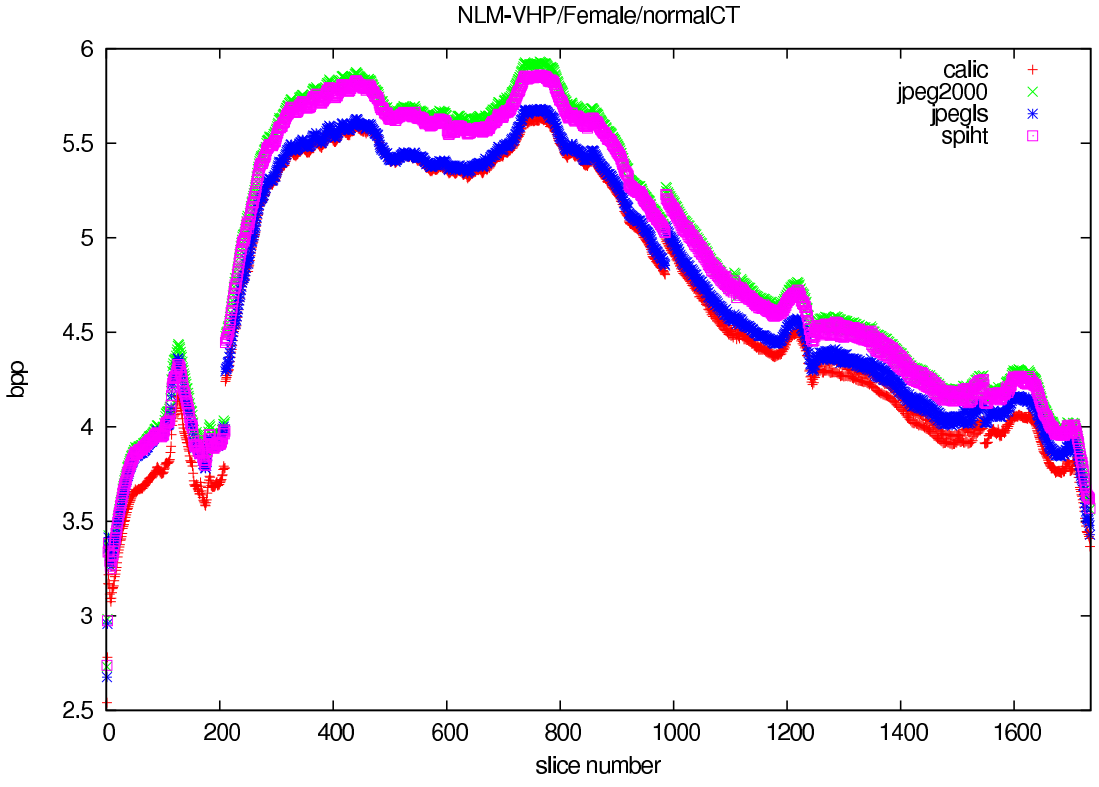


FIG. A.3 – volume (C), CT avec un bruit très marqué

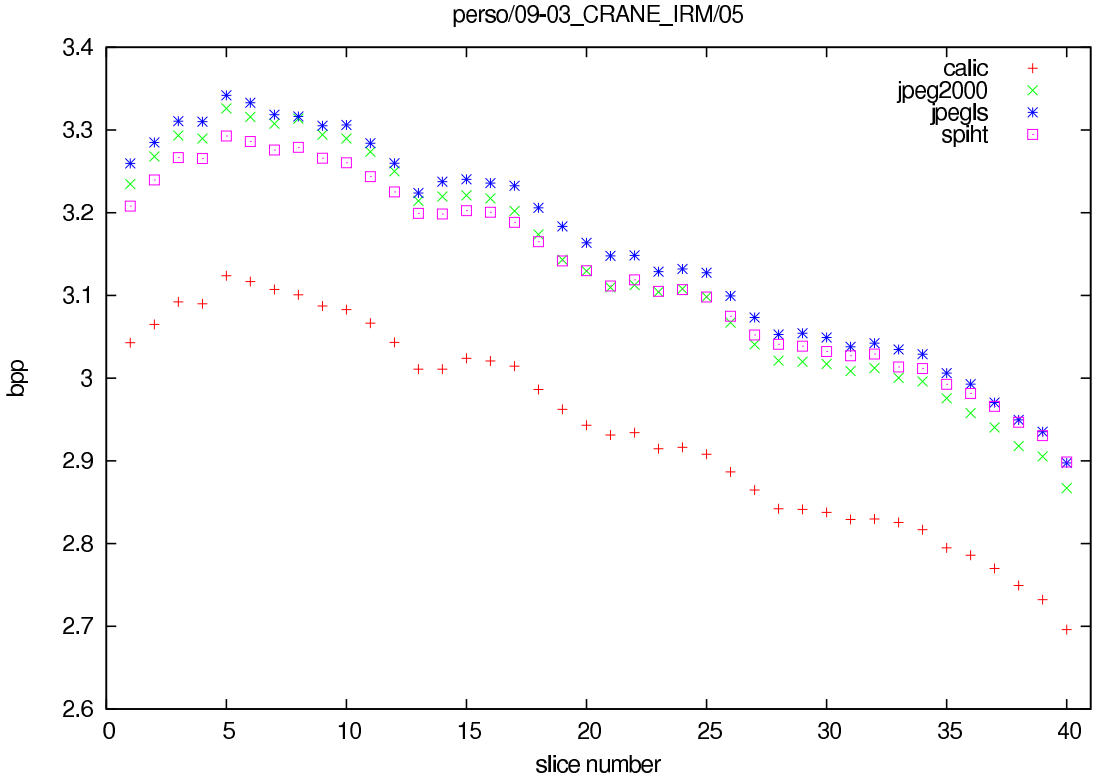


FIG. A.4 – volume (D), IRM

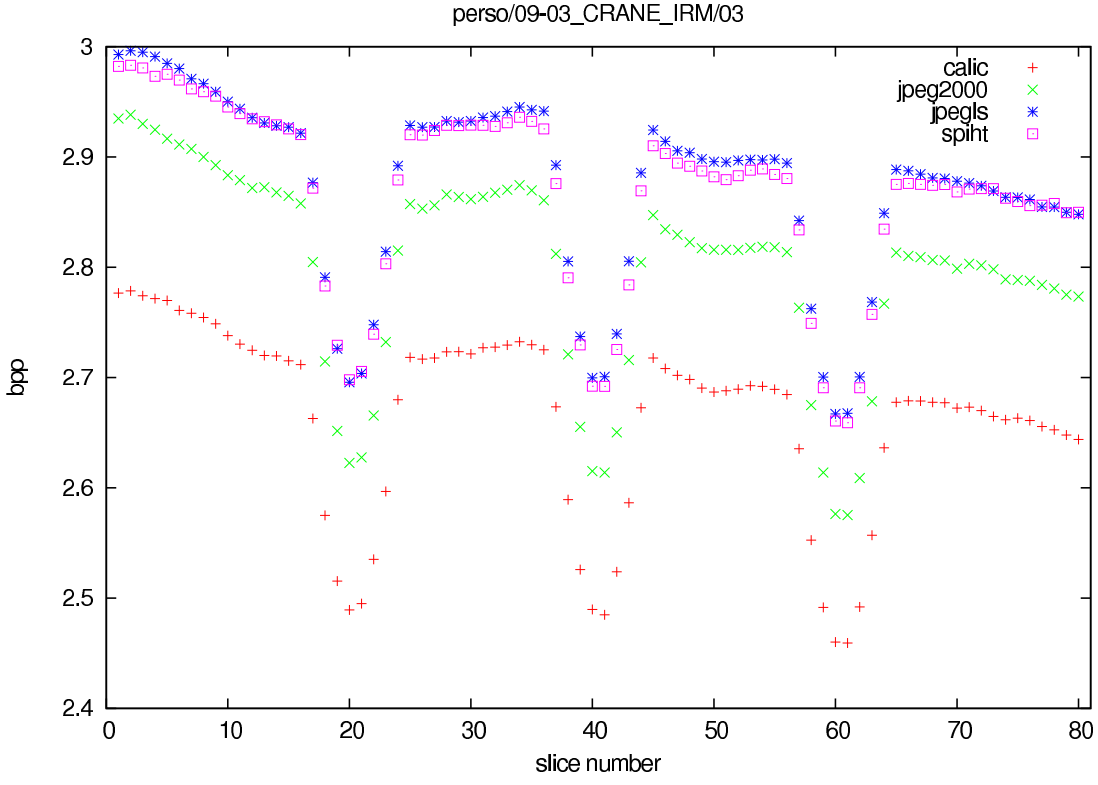


FIG. A.5 – volume (E), IRM3D

Annexe B

Compression Intra/Inter

Cette annexe présente les courbes des résultats de la compression coupe par coupe et avec décorrélation inter-coupe (cf. section 6.2) des cinq volumes présentés dans la section 6.4. Ces cinq volumes ont été sélectionnés parmi un ensemble de plus de 100 volumes en provenance de différentes bases d'images (cf. section 1.4) sur lesquels ont portés nos investigations.

En abscisses les numéros de coupes et en ordonnées le nombre de bits par pixels utilisés après compression.

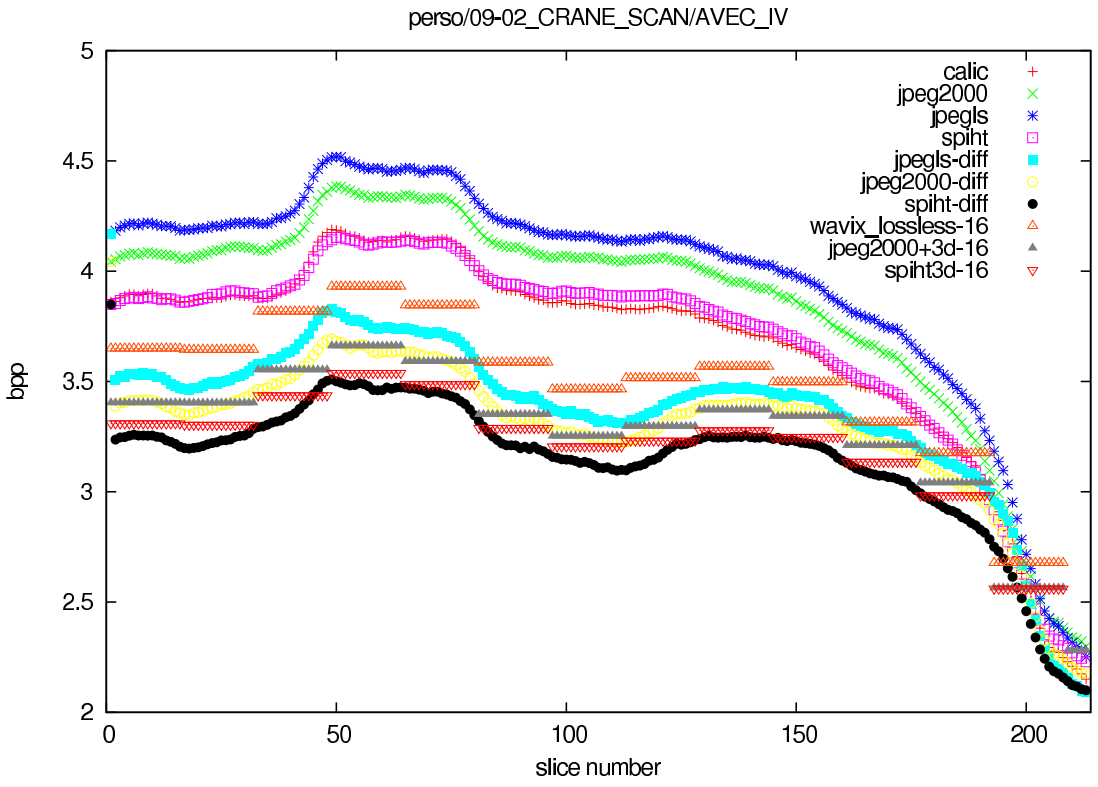


FIG. B.1 – volume (A), CT avec un bruit faible

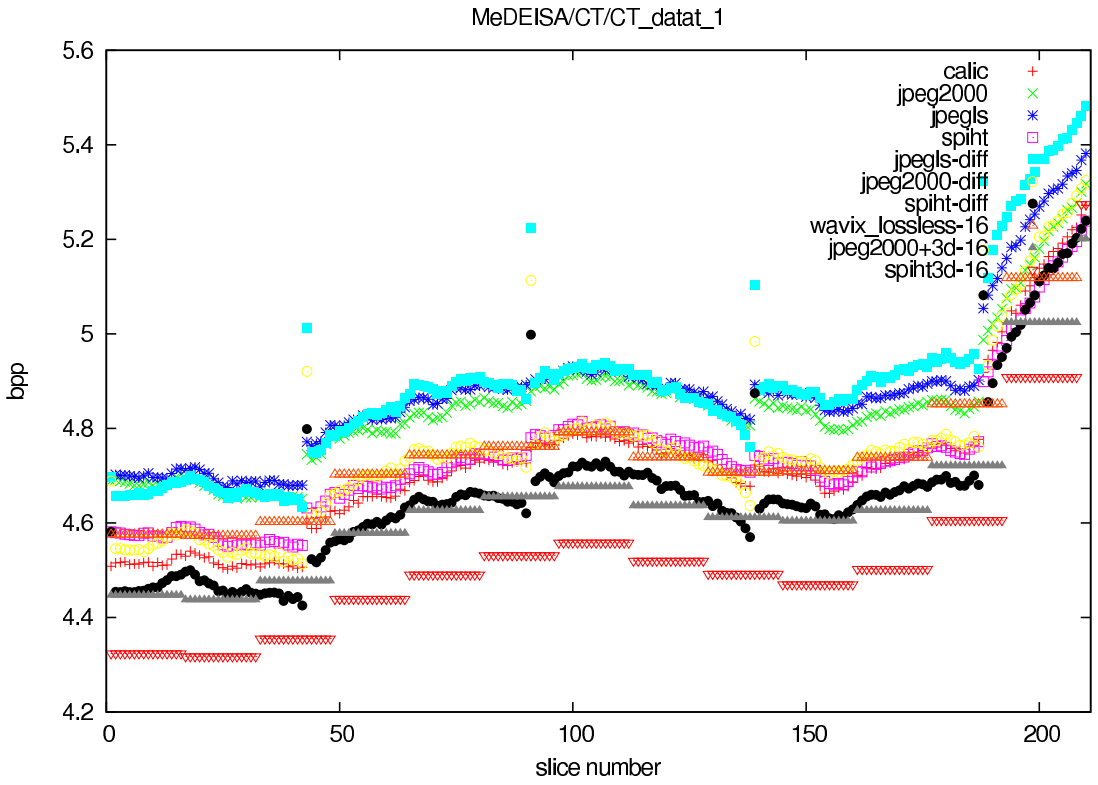


FIG. B.2 – volume (B), CT avec un bruit de reconstruction important mais diffus (filtrage)

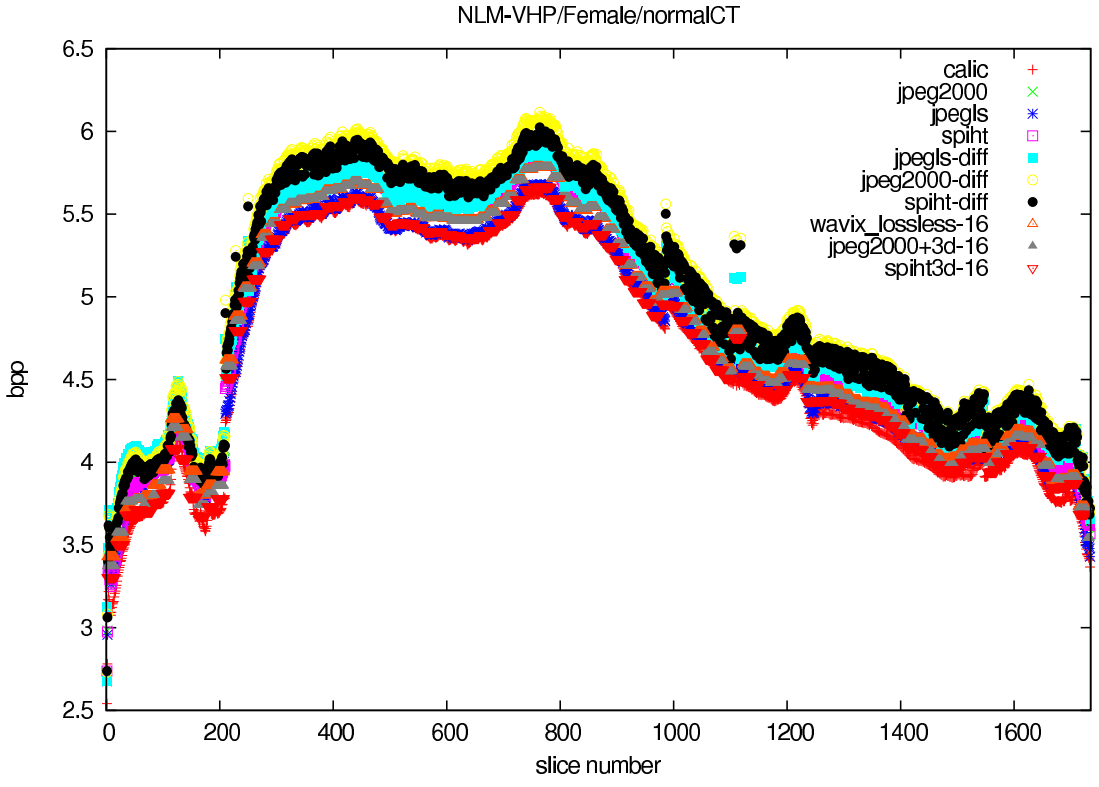


FIG. B.3 – volume (C), CT avec un bruit très marqué

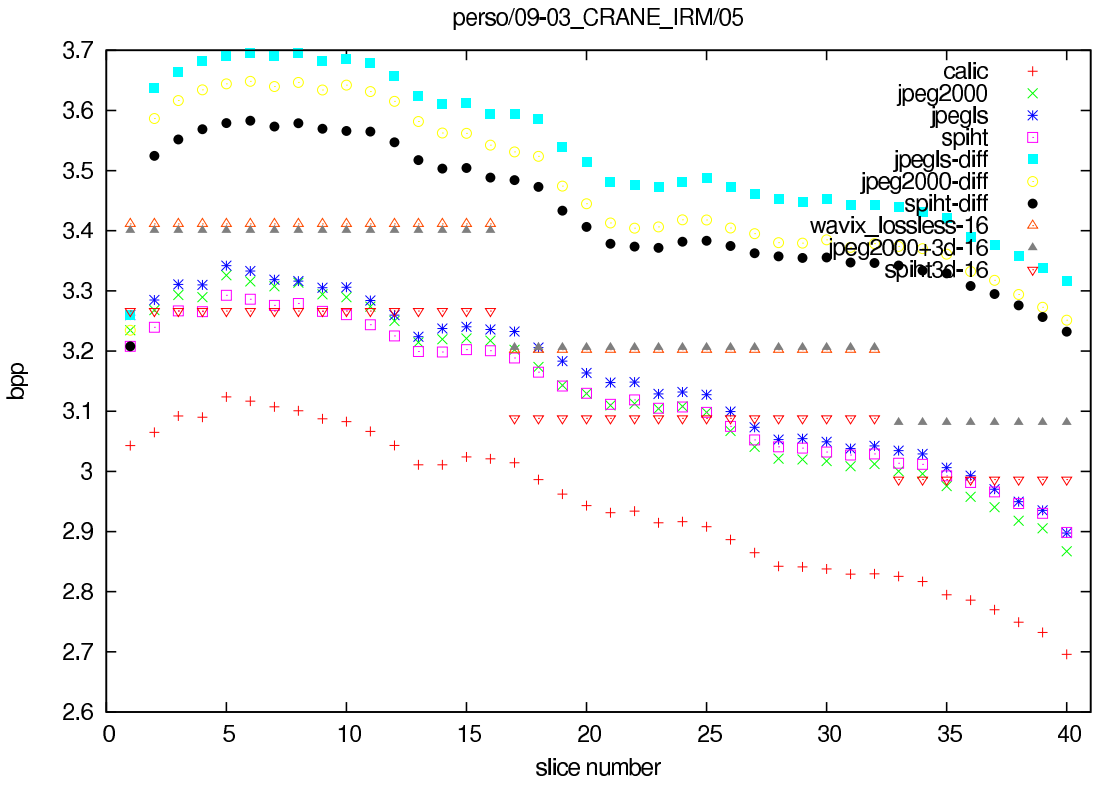


FIG. B.4 – volume (D), IRM

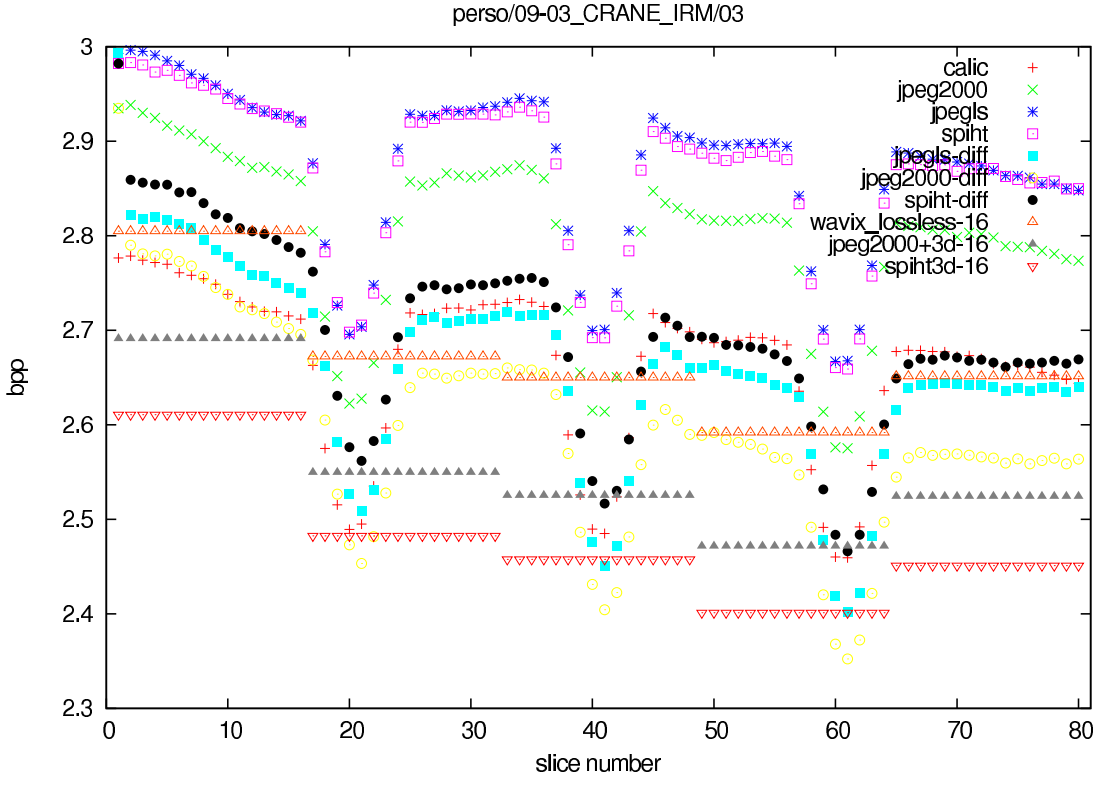


FIG. B.5 – volume (E), IRM3D

Annexe C

Compression ROI

Cette annexe présente les courbes des résultats de la compression coupe par coupe et avec décorrélation inter-coupe (cf. section 6.2) de quatre des cinq volumes présentés dans la section 6.4 après suppression de l'information hors de la région d'intérêt définie par l'approche de la section 6.3. Les résultats sur le volume (B) ne sont pas présentés : la région d'intérêt englobant quasiment la totalité de l'information reconstruite, les taux de compression sont presque identiques à ceux des annexes précédentes.

En abscisses les numéros de coupes et en ordonnées le nombre de bits par pixels utilisés après compression.

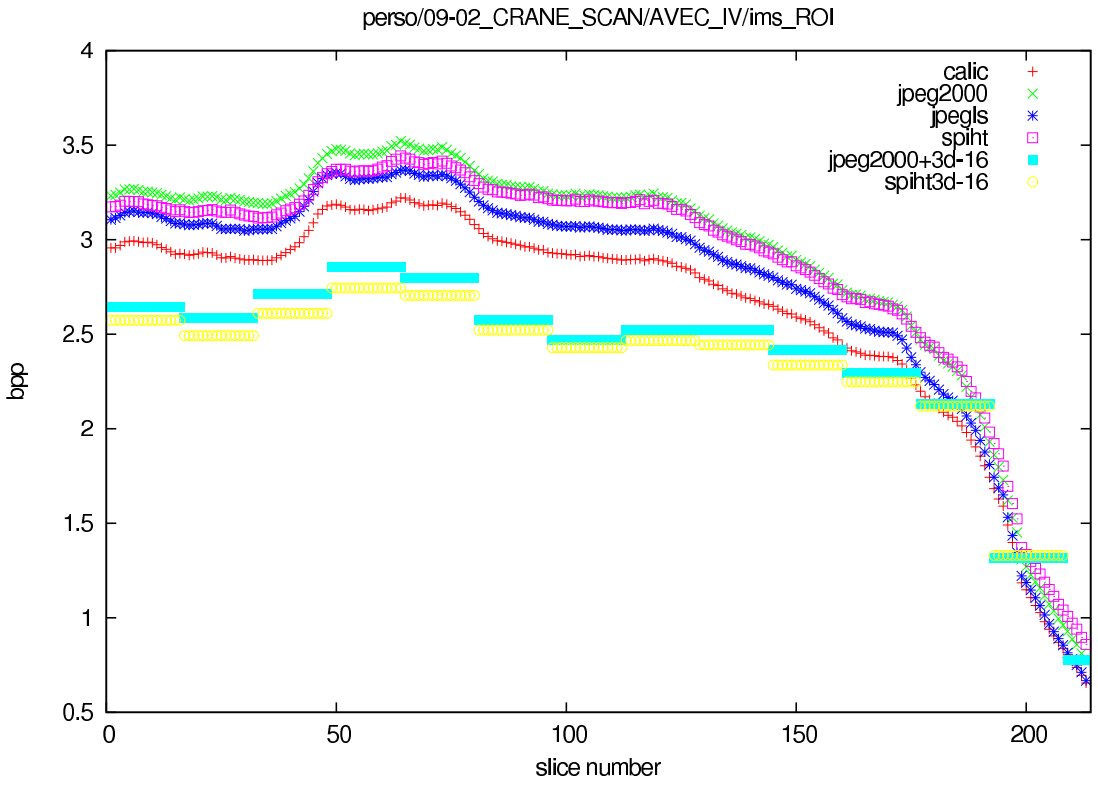


FIG. C.1 – volume (A), CT avec un bruit faible

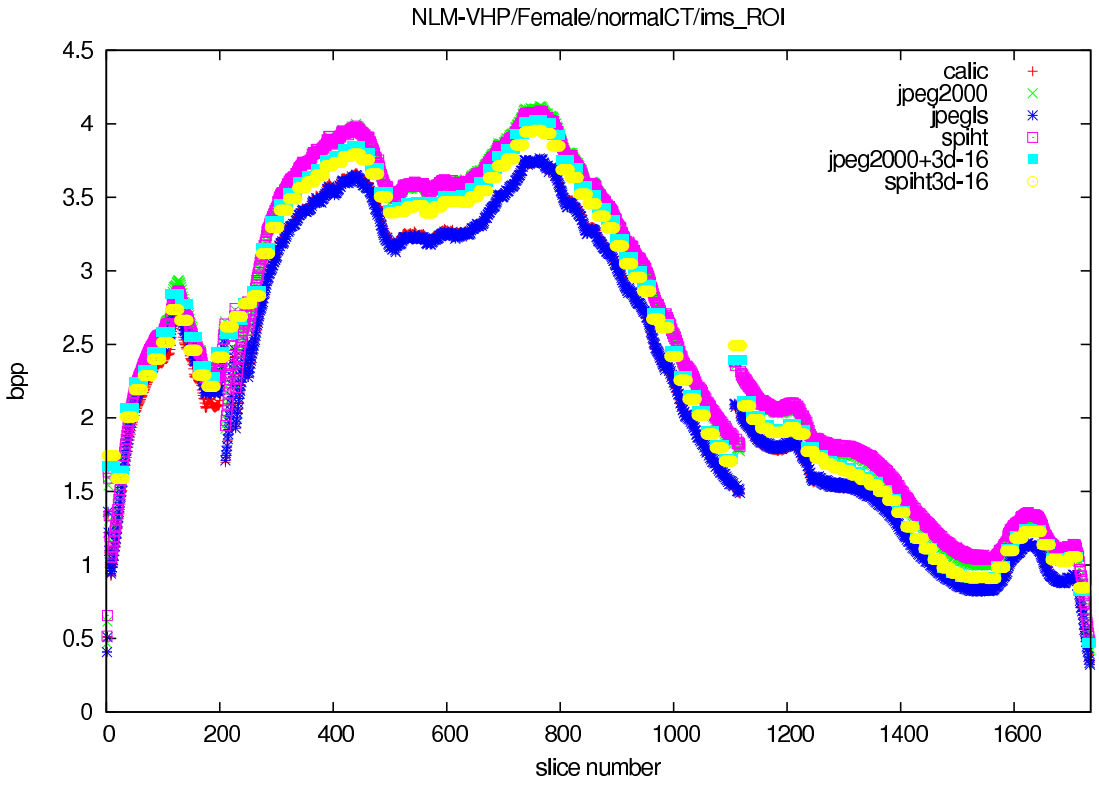


FIG. C.2 – volume (C), CT avec un bruit très marqué

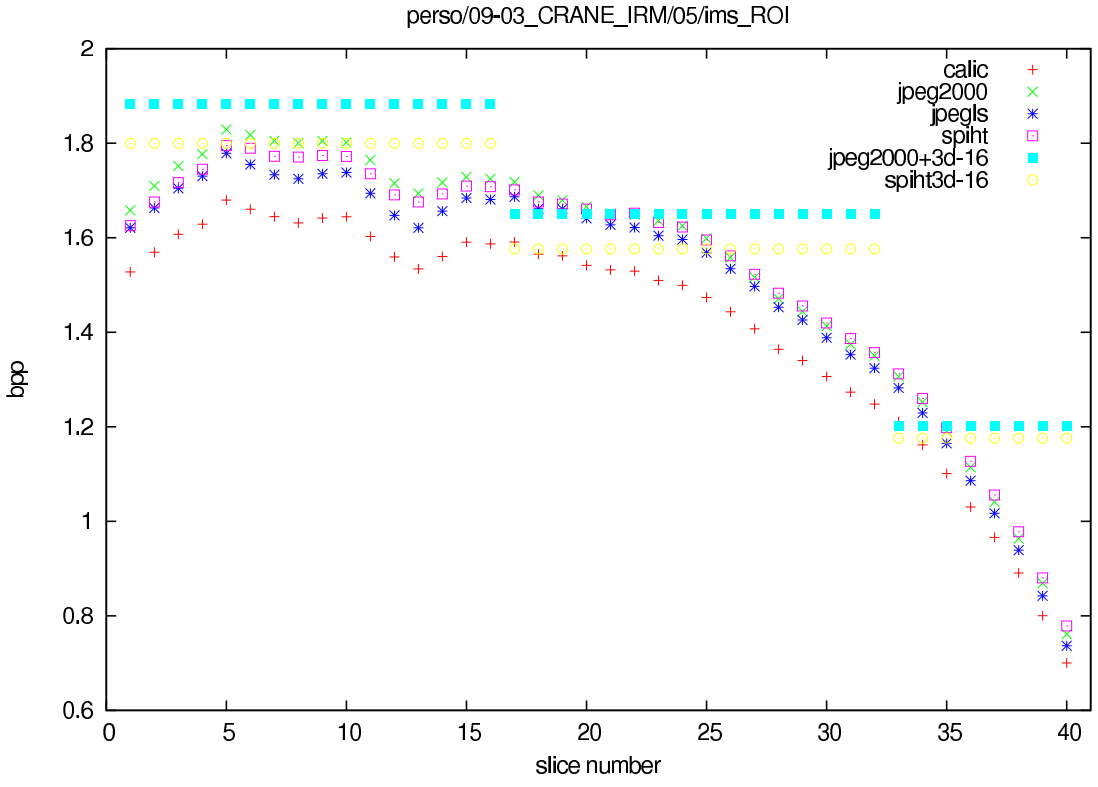


FIG. C.3 – volume (D), IRM

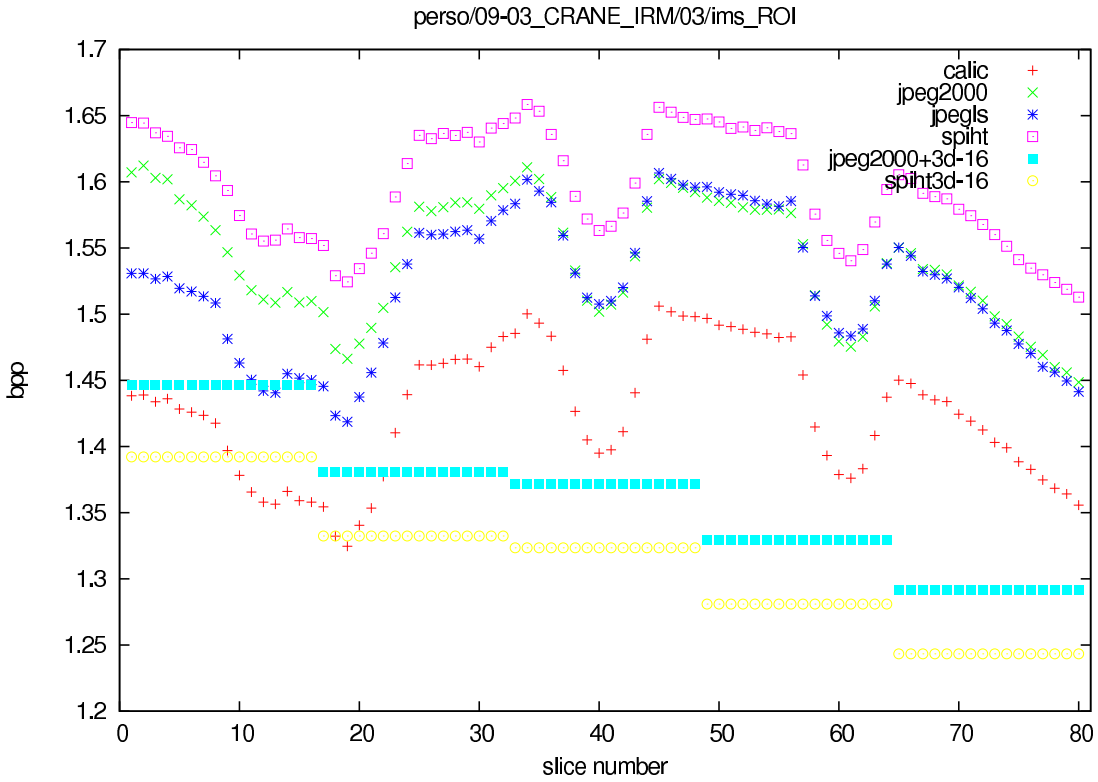


FIG. C.4 – volume (E), IRM3D

Bibliographie

[AAB97] Andrea Abrardo, Luciano Alparone, and Franco Bartolini. Encoding-interleaved hierarchical interpolation for lossless image compression. *Signal Processing*, 56(3) :321 – 328, 1997.

[AAB02] B. Aiazzi, L. Alparone, and S. Baronti. Fuzzy logic-based matching pursuits for lossless predictive coding of still images. *Fuzzy Systems, IEEE Transactions on*, 10(4) :473–483, Aug 2002.

[AABA96] B. Aiazzi, P.S. Alba, S. Baronti, and L. Alparone. Three-dimensional lossless compression based on a separable generalized recursive interpolation. In *International Conference on Image Processing (ICIP) 1996*, volume 1, pages 85–88 vol.1, Sep 1996.

[AABL97] B. Aiazzi, L. Alparone, S. Baronti, and F. Lotti. Lossless image compression by quantization feedback in a content-driven enhanced laplacian pyramid. *Image Processing, IEEE Transactions on*, 6(6) :831–843, Jun 1997.

[ABA01] B. Aiazzi, S. Baronti, and L. Alparone. Near-lossless compression of coherent image data. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 490–493 vol.3, 2001.

[Abh03] G.C.K. Abhayaratne. Modifying integer wavelet transforms for scalable near-lossless image coding. In *Visual Communications and Image Processing (VCIP) 2003*, volume Proc. SPIE 5150, pages 1697–1708, 2003.

[Abh07] G. Charith K. Abhayaratne. Reversible integer-to-integer mapping of n-point orthonormal block transforms. *Signal Processing*, 87(5) :950–969, 2007.

[Ada05] Michael D. Adams. The JPEG-2000 still image compression standard. *Revision of the JPEG-2000 tutorial appeared in JPEG working group document WG1N1734*, December 03 2005.

[AHS02] A. Abu-Hajar and R. Sankar. Integer-to-integer shape adaptive wavelet transform for region of interest image coding. In *Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop. Proceedings of 2002 IEEE 10th*, pages 94–97, Oct. 2002.

[AHS04] A. Abu-Hajar and R. Sankar. Region of interest coding using partial-spiht. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, pages iii–657–60 vol.3, May 2004.

[And97] J. Andrew. A simple and efficient hierarchical image coder. In *Image Processing, 1997. Proceedings., International Conference on*, volume 3, pages 658–661 vol.3, Oct 1997.

[ANR74] N. Ahmed, T. Natarajan, and K.R. Rao. Discret cosine transform. *IEEE Signal Processing Magazine*, 14(2) :24–41, 1974.

[BA83] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4) :532–540, Apr 1983.

[BCK07] Eric Bodden, Malte Clasen, and Joachim Kneis. Arithmetic coding revealed - a guided tour from theory to praxis. Technical Report SABLE-TR-2007-5, Sable Research Group, School of Computer Science, McGill University, Montréal, Québec, Canada, May 2007. http ://www.bodden.de/legacy/arithmetic-coding/.

[BdP96] J. Browne and A.B. de Pierro. A row-action alternative to the em algorithm for maximizing likelihood in emission tomography. *Medical Imaging, IEEE Transactions on*, 15(5) :687–699, Oct 1996.

[BDR05] M. Babel, O. Deforges, and J. Ronsin. Interleaved s+p pyramidal decomposition with refined prediction model. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 2 :II–750–3, Sept. 2005.

[BFG05] G. Boisson, E. François, and C. Guillemot. Twavix : une solution basée-ondelettes (t+2d) pour le codage vidéo scalable. In *Actes de la 10ème conférence sur la COmpression et la REprésentation des Signaux Audiovisuels, CORESA 2005*, Rennes, France, nov 2005.

[BRS+04] M. Bertram, G. Rose, D. Schafer, J. Wiegert, and T. Aach. Directional interpolation of sparsely sampled cone-beam ct sinogram data. In *Biomedical Imaging : Nano to Macro, 2004. IEEE International Symposium on*, pages 928–931 Vol. 1, April 2004.

[BSS09] Ian Blanes and Joan Serra-Sagristà. Clustered reversible-klt for progressive lossy-to-lossless 3d image coding. *Data Compression Conference*, 0 :233–242, 2009.

[CAR08] L'Association canadienne des radiologistes CAR. Normes de la CAR en matière de compression irréversible pour l'imagerie numérique diagnostique en radiologie, 2008.

[CDF90] A. Cohen, Ingrid Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communication on Pure and Applied Mathematics*, 45(5) :485–560, 1990.

[Che07] Yao-Tien Chen. *Medical Image Compression and Segmentation based on Statistical Inferences*. PhD thesis, Institute of Computer Science and Information Engineering from National Central University, Taiwan, R.O.C., May 2007.

[CKP04] Sungdae Cho, Dongyoun Kim, and William A. Pearlman. Lossless compression of volumetric medical images with improved three-dimensional spiht algorithm. *Journal of Digital Imaging*, 17(1) :57–63, March 2004.

[Clu00] David A. Clunie. Lossless compression of grayscale medical images - effectiveness of traditional and state of the art approaches. In *Medical Imaging 2000 : PACS Design and Evaluation : Engineering and Clinical Issues*, volume 3980 of *Proceedings of SPIE*, pages 74–84, San Diego, Calif, USA, February 2000.

[CPS05] Yushin Cho, W.A. Pearlman, and A. Said. Low complexity resolution progressive image coding algorithm : progres (progressive resolution decompression). *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 3 :III–49–52, Sept. 2005.

[CTC06] Y.T. Chen, D.C. Tseng, and P.C. Chang. Wavelet-based image compression with polygon-shaped region of interest. In *2006 IEEE Pacific-Rim Symposium on Image and Video Technology (PSIVT'06)*, pages 878–887, 2006.

[Dal93] Scott Daly. The visible differences predictor : an algorithm for the assessment of image fidelity. In Watson, editor, *Digital images and human vision*, pages 179–206. MIT Press, Cambridge, MA, USA, 1993.

[Dal94] S. Daly. A visual model for optimizing the design of image processing algorithms. *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, 2 :16–20 vol.2, Nov 1994.

[Dau88] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communication on Pure and Applied Mathematics*, 41(7) :909–996, 1988.

[Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets (C B M S - N S F Regional Conference Series in Applied Mathematics)*. Soc for Industrial & Applied Math, December 1992.

[DBBR07] O. Deforges, M. Babel, L. Bedat, and J. Ronsin. Color lar codec : A color image representation and compression scheme based on local resolution adjustment and self-extracting region representation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(8) :974–987, Aug. 2007.

[DBM06] Olivier Déforges, Marie Babel, and Jean Motsch. The RWHT+P for an improved lossless multiresolution coding. In *EUSIPCO'06*, Florence, Italy, Sept. 2006.

[DGSWAL05] M. Díez-García, F. Simmross-Wattenberg, and C. Alberola-López. A lossless compression algorithm based on predictive coding for volumetric medical datasets. In *European Signal Processing Conference, EUSIPCO-05*, Antalya, Turkey, Sept. 2005.

[dRP99] D. de Rycke and W. Philips. Lossless non-linear predictice coding of video data through context matching. In M. Torres, B. Sanchez, and D. Langlois, editors, *The 5th International Conference on Information Systems Analysis and Synthesis (ISAS ?99)*, volume 6, pages 42–49, Orlando, USA, Aug. 1999.

[DRPV06] C. Delgorge, C. Rosenberger, G. Poisson, and P. Vieyres. Towards a new tool for the evaluation of the quality of ultrasound compressed images. *Medical Imaging, IEEE Transactions on*, 25(11) :1502–1509, Nov. 2006.

[dSS08] Rafael A.P. dos Santos and Jacob Scharcanski. Lossless and near-lossless digital angiography coding using a two-stage motion compensation approach. *Computerized Medical Imaging and Graphics*, 32(5) :379–387, 2008.

[DvAPL97] Koen Denecker, Steven van Assche, Wilfried Philips, and Ignace Lemahieu. State of the art concerning lossless medical image coding. In *in Proceedings of the PRORISC IEEE Benelux Workshop on Circuits, Systems and Signal Processing*, pages 129–136, 1997.

[EC91] W.H.R. Equitz and T.M. Cover. Successive refinement of information. *Information Theory, IEEE Transactions on*, 37(2) :269–275, Mar 1991.

[ELK95] O. Egger, Wei Li, and M. Kunt. High compression image coding using an adaptive morphological subband decomposition. *Proceedings of the IEEE*, 83(2) :272–287, Feb 1995.

[Gau06] Yann Gaudeau. *Contributions en compression d'images médicales 3D et d'images naturelles 2D*. PhD thesis, Université Henri Poincaré, Nancy 1, 2006.

[GBH70] R. Gordon, R. Bender, and G. T. Herman. Algebraic reconstruction techniques (art) for three dimensional electron microscopy and x-ray photography. *Journal of Theoretical Biology*, 29(3) :471–481, 1970.

[GEVK00] R. Gruter, O. Egger, J.M. Vesin, and M. Kunt. Rank-order polynomial subband decomposition for medical image compression. *Medical Imaging, IEEE Transactions on*, 19(10) :1044–1052, Oct. 2000.

[GG91] Allen Gersho and Robert M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[Gil72] Peter Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *Journal of Theoretical Biology*, 36(1) :105 – 117, 1972.

[GM09] Y. GAUDEAU and J.-M. MOUREAUX. Lossy compression of volumetric medical images with 3d dead zone lattice vector quantization. *Annals of telecommunications*, 64(5-6), may 2009.

[Gol66] S. W. Golomb. Run length encodings. In *IEEE Transactions on Information Theory IT-12*, pages 399–401, July 1966.

[GSP05] N. Gupta, M.N.S. Swamy, and E. Plotkin. Despeckling of medical ultrasound images using data and rate adaptive lossy compression. *Medical Imaging, IEEE Transactions on*, 24(6) :743–754, June 2005.

[HCL03] Wen-Jyi Hwang, Ching-Fung Chine, and Kuo-Jung Li. Scalable medical data compression and transmission using wavelet transform for telemedicine applications. *Information Technology in Biomedicine, IEEE Transactions on*, 7(1) :54–63, March 2003.

[HL94] H.M. Hudson and R.S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *Medical Imaging, IEEE Transactions on*, 13(4) :601–609, Dec 1994.

[HLL75] Gabor T. Herman, Arnold Lent, and Peter H. Lutz. Iterative relaxation methods for image reconstruction. In *ACM 75 : Proceedings of the 1975 annual conference*, pages 169–174, New York, NY, USA, 1975. ACM.

[HMa] Denis Hoa and Antoine Micheau. e-MRI : MRI physics interactive tutorial. web site : http://www.imaios.com/en/e-Courses/e-MRI.

[HMb] Denis Hoa and Antoine Micheau. L'IRM pas à pas, cours interactif sur l'imagerie par résonance magnétique. site web : http://www.imaios.com/fr/e-Cours/e-MRI.

[HR73] G.T. Herman and S.W. Rowland. Three methods for reconstructing objects from x rays : A comparative study. *CGIP*, 2(2) :151–178, October 1973.

[Hsi01] Shih-Ta Hsiang. Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling. *Data Compression Conference, 2001. Proceedings. DCC 2001.*, pages 83–92, 2001.

[Huf52] D.A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9) :1098–1101, Sept. 1952.

[HW01] Shih-Ta Hsiang and John W. Woods. Embedded video coding using invertible motion compensated 3-d subband/wavelet filter bank. *Signal Processing : Image Communication*, 16(8) :705 – 724, 2001.

[HW02] Shih-Ta Hsiang and John W. Woods. Highly scalable and perceptually tuned embedded subband/wavelet image coding. In *VCIP*, pages 1153–1164, 2002.

[ITU05] ITU-T Recommendation T.851. ITU-T T.81 (JPEG-1)-based still-image coding using an alternative arithmetic coder. Technical report, International Telecommunication Union, September 2005.

[KA08] Andrew Kingston and Florent Autrusseau. Lossless image compression via predictive coding of discrete radon projections. *Signal Processing : Image Communication*, 23 :313–324, 2008.

[KBB+08] D. Koff, P. Bak, P. Brownrigg, D. Hosseinzadeh, A. Khademi, A. Kiss, L. Lepanto, T. Michalak, H. Shulman, and A. Volkening. Pan-canadian evaluation of irreversible compression ratios ("lossy" compression) for development of national guidelines. *Journal of Digital Imaging*, 2008.

[KJML05] Xie Kai, Yang Jie, Zhu Yue Min, and Li Xiao Liang. Hvs-based medical image compression. *European Journal of Radiology*, 55(1) :139–145, July 2005.

[KL05] Lih-Jen Kau and Yuan-Pei Lin. Adaptive lossless image coding using least squares optimization with edge-look-ahead. *Circuits and Systems II : Express Briefs, IEEE Transactions on*, 52(11) :751–755, Nov. 2005.

[KOK+98] Juha Kivijarvi, Tiina Ojala, Timo Kaukoranta, Attila Kuba, László G. Nyúl, and Olli Nevalainen. A comparison of lossless compression methods for medical images. *Computerized Medical Imaging and Graphics*, 22 :323–339, 1998.

[LBG80] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *Communications, IEEE Transactions on*, 28(1) :84–95, Jan 1980.

[LBL87] Kenneth Lange, Mark Bahn, and Roderick Little. A theoretical study of some maximum likelihood algorithms for emission and transmission tomography. *Medical Imaging, IEEE Transactions on*, 6(2) :106–114, June 1987.

[LF97] J. Lubin and D. Fibush. Sarnoff JND vision model. Technical report, T1A1.5 Working Group Document #97-612, T1 Standards Committee, 1997.

[LGT88] D. Le Gall and A. Tabatabai. Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 761–764 vol.2, Apr 1988.

[LL00] Shipeng Li and Weiping Li. Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 10(5) :725–743, Aug 2000.

[LLF03] S.-C.B. Lo, Huai Li, and M.T. Freedman. Optimization of wavelet decomposition for image compression and feature preservation. *Medical Imaging, IEEE Transactions on*, 22(9) :1141–1151, Sept. 2003.

[LLK+05] Kyoung Ho Lee, Hak Jong Lee, Jae Hyoung Kim, Heung Sik Kang, Kyung Won Lee, Helen Hong, Ho Jun Chin, and Kyoo Seob Ha. Managing the ct data explosion : Initial experiences of archiving volumetric datasets in a mini-pacs. *Journal of Digital Imaging*, 18(3) :188–195, Sept. 2005.

[Llo82] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2) :129–137, Mar 1982.

[LO01] Xin Li and Michael T. Orchard. Edge-directed prediction for lossless compression of natural images. *IEEE Transactions on Image Processing*, 10 :813–817, 2001.

[Lub93] Jeffrey Lubin. The use of psychophysical data and models in the analysis of display system performance. In Watson, editor, *Digital images and human vision*, pages 163–178. MIT Press, Cambridge, MA, USA, 1993.

[Lub95] Jeffrey Lubin. A visual discrimination model for imaging system design and evaluation. In E. Peli, editor, *Vision Models for Target Detection and Recognition*, volume 2 of *Series on Information Display*, pages 245–283. World Scientific Publishing, 1995.

[Lub97] J. Lubin. A human vision system model for objective picture quality measurements. *Broadcasting Convention, 1997. International*, pages 498–503, Sep 1997.

[Mal06] H.S. Malvar. Adaptive run-length/golomb-rice encoding of quantized generalized gaussian sources with unknown statistics. *Data Compression Conference, 2006. DCC 2006. Proceedings*, pages 23–32, March 2006.

[Mal08] Stéphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition : The Sparse Way*. Academic Press, 3 edition, December 2008.

[Max60] J. Max. Quantizing for minimum distortion. *Information Theory, IRE Transactions on*, 6(1) :7–12, March 1960.

[MC04] Shaou-Gang Miaou and Shih-Tse Chen. Automatic quality control for wavelet-based compression of volumetric medical images using distortion-constrained adaptive vector quantization. *Medical Imaging, IEEE Transactions on*, 23(11) :1417–1429, Nov. 2004.

[MCC99] A. Munteanu, J. Cornelis, and P. Cristea. Wavelet-based lossless compression of coronary angiographic images. *Medical Imaging, IEEE Transactions on*, 18(3) :272–281, March 1999.

[Men00] Gloria Menegaz. *Model-based coding of multi-dimensional data with applications to medical imaging*. PhD thesis, école polytechnique fédérale de Lausanne, 2000.

[MK06] A.A. Moinuddin and E. Khan. Wavelet based embedded image coding using unified zero-block-zero-tree approach. *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 2 :II–II, May 2006.

[MMI00] I. Matsuda, H. Mori, and S. Itoh. Lossless coding of still images using minimum-rate predictors. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 1, pages 132–135 vol.1, 2000.

[MOUI05] Ichiro Matsuda, Nau Ozaki, Yuji Umezu, and Susumu Itoh. Lossless coding using variable block-size adaptive prediction optimized for each image. In *13th European Signal Processing Conference (EUSIPCO 2005)*, Sep. 2005.

[MT97] B. Meyer and P. Tischer. TMW - a New Method for Lossless Image Compression. In *International Picture Coding Symposium PCS97*, sept. 1997.

[MT98] B. Meyer and P. Tischer. Extending tmw for near lossless compression of greyscale images. In *Data Compression Conference, 1998. DCC '98. Proceedings*, pages 458–470, Mar-1 Apr 1998.

[MT01] B. Meyer and P. Tischer. Glicbawls - grey level image compression by adaptive weighted least squares. In *Proc. Data Compression Conference*, page 503, Snowbird, Utah, USA, Mar 2001. IEEE Computer Society.

[MT02] G. Menegaz and J.-P. Thiran. Lossy to lossless object-based coding of 3-d mri data. *Image Processing, IEEE Transactions on*, 11(9) :1053–1061, Sep 2002.

[MT03] G. Menegaz and J.-P. Thiran. Three-dimensional encoding/two-dimensional decoding of medical data. *Medical Imaging, IEEE Transactions on*, 22(3) :424–440, March 2003.

[MYRP07] Byungjun Min, Sook Yoon, Jungjin Ra, and Dong Sun Park. Enhanced renormalization algorithm in mq-coder of jpeg2000. *Information Technology Convergence, 2007. ISITC 2007. International Symposium on*, pages 213–216, Nov. 2007.

[NACM07] Amine Naït-Ali and Christine Cavaro-Ménard, editors. *Compression des images et des signaux médicaux*. LAVOISIER, 2007.

[NACM08] Amine Naït-Ali and Christine Cavaro-Ménard, editors. *Compression of Biomedical Images and Signals*. ISTE / WILEY, 2008.

[NMOL96] A. Nosratinia, N. Mohsenian, M.T. Orchard, and B. Liu. Interframe coding of magnetic resonance images. *Medical Imaging, IEEE Transactions on*, 15(5) :639–647, Oct 1996.

[OBBO06] P.M.A. van Ooijen, P.J.M. ten Bhomer, A. Broekema, and M. Oudkerk. Shifting storage requirements due to modality changes in six years of pacs. In *The 24th International EuroPACS Conference*, Trondheim, Norway, June 15th to 17th 2006.

[OBO05] P.M.A. van Ooijen, P.J.M. ten Bhomer, and M. Oudkerk. Pacs storage requirements-influence of changes in imaging modalities. *International Congress Series 1281*, pages 888–893, 2005.

[PBDB08] François Pasteau, Marie Babel, Olivier Déforges, and Laurent Bédat. Interleaved s+p scalable coding with inter-coefficient classification methods. In *Proc. 16th European Signal Processing Conference*, Lausanne, Switzerland, August 25-29 2008.

[PINS04] W.A. Pearlman, A. Islam, N. Nagaraj, and A. Said. Efficient, low-complexity image coding with a set-partitioning embedded block coder. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(11) :1219–1235, Nov. 2004.

[PJB87] J. Princen, A. Johnson, and A. Bradley. Subband/transform coding using filter bank designs based on time domain aliasing cancellation. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.*, volume 12, pages 2161–2164, Apr 1987.

[PMLA88] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, Jr., and R. B. Arps. An overview of the basic principles of the q-coder adaptive binary arithmetic coder. *IBM J. Res. Develop.*, 32(6) :717–726, 1988.

[PPT+03] M. Penedo, W.A. Pearlman, P.G. Tahoces, M. Souto, and J.J. Vidal. Region-based wavelet coding methods for digital mammography. *Medical Imaging, IEEE Transactions on*, 22(10) :1288–1296, Oct. 2003.

[PR07] B. Prabhakar and M. Ramasubba Reddy. Hvs scheme for dicom image compression : Design and comparative performance evaluation. *European Journal of Radiology*, 63(Issue 1) :128–135, July 2007.

[PTMO07] B. Penna, T. Tillo, E. Magli, and G. Olmo. Transform coding techniques for lossy hyperspectral data compression. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(5) :1408–1421, May 2007.

[PvAdRD01] W. Philips, S. van Assche, D. de Rycke, and K. Denecker. State-of-the-art techniques for lossless compression of 3d medical image sets. *Computerized Medical Imaging and Graphics*, 25(2) :173–185, 3 2001.

[Ric03] Iain Richardson. *H.264 and MPEG-4 video compression*. WILEY, 2003.

[RV93] P. Roos and M.A. Viergever. Reversible 3-D decorrelation of medical images. *Medical Imaging, IEEE Transactions on*, 12(3) :413–420, Sep 1993.

[RVvDP88] P. Roos, M.A. Viergever, M.C.A. van Dijke, and J.H. Peters. Reversible intraframe compression of medical images. *Medical Imaging, IEEE Transactions on*, 7(4) :328–336, Dec 1988.

[Sai04] Amir Said. Introduction to arithmetic coding - theory and practice. Technical Report HPL-2004-76, Imaging Systems Laboratory, HP Laboratories Palo Alto, April 2004. http ://www.hpl.hp.com/techreports/2004/HPL-2004-76.pdf.

[SCE01] Athanassios Skodras, Charilaos Chritopolos, and Touradj Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, pages 36–58, September 2001.

[Sha48] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 :379–423, 623–656, July,October 1948.

[Sha93] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *Signal Processing, IEEE Transactions on*, 41(12) :3445–3462, Dec 1993.

[SMB+03] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis. Wavelet coding of volumetric medical datasets. *Medical Imaging, IEEE Transactions on*, 22(3) :441–458, March 2003.

[SP93] Amir Said and William A. Pearlman. Reversible image compression via multiresolution representation and predictive coding. In *Visual Communications and Image Processing*, number 2094 in SPIE, pages 664–674, Nov. 1993.

[SP96a] A. Said and W.A. Pearlman. An image multiresolution representation for lossless and lossy compression. *Image Processing, IEEE Transactions on*, 5(9) :1303–1310, Sep 1996.

[SP96b] A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(3) :243–250, Jun 1996.

[SP97] Amir Said and William A. Pearlman. Low-complexity waveform coding via alphabet and sample-set partitioning. In *SPIE Visual Communications and Image Processing '97*, volume 3024, pages 25–37, Feb. 1997.

[SR05] R. Srikanth and A.G. Ramakrishnan. Contextual encoding in uniform and adaptive mesh-based lossless compression of mr images. *Medical Imaging, IEEE Transactions on*, 24(9) :1199–1206, Sept. 2005.

[SV82] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *Medical Imaging, IEEE Transactions on*, 1(2) :113–122, Oct. 1982.

[Swe96] W. Sweldens. Wavelets and the lifting scheme : A 5 minute tour. *Z. Angew. Math. Mech.*, 76 (Suppl. 2) :41–44, 1996.

[Tau00] D. Taubman. High performance scalable image compression with ebcot. *Image Processing, IEEE Transactions on*, 9(7) :1158–1170, Jul 2000.

[TK03] E Tanaka and H Kudo. Subset-dependent relaxation in block-iterative algorithms for image reconstruction in emission tomography. *Physics in Medicine and Biology*, 48(10) :1405–1422, 2003.

[TM01] David Taubman and Michael Marcellin. *JPEG2000 : Image Compression Fundamentals, Standards and Practice*. Springer, 2001.

[TSS+08] Chengjie Tu, Sridhar Srinivasan, Gary J. Sullivan, Shankar Regunathan, and Henrique S. Malvar. Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR / HD Photo. In Andrew G. Tescher, editor, *Applications of Digital Image Processing XXXI*, volume 7073, pages 70730C–1–70730C–12. SPIE, 2008.

[US07] G. Ulacha and R. Stasinski. Texture matching method for lossless image coding. *Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop on*, pages 130–132, June 2007.

[US08] G. Ulacha and R. Stasinski. A new simple context lossless image coding algorithm based on adaptive context arithmetic coder. *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pages 45–48, June 2008.

[vAdRPL99] Steven van Assche, Dirk de Rycke, Wilfried Philips, and Ignace Lemahieu. Exploiting interframe redundancies in the lossless compression of 3d medical images. In *STW, Program for Research on Integrated Systems and Circuits (ProRISC) 99*, pages 521–527, 1999.

[VGP02] J. Viéron, C. Guillemot, and S. Pateux. Motion compensated 2d+t wavelet analysis for low rate fgs video compression. In *of the International Thyrrhenian workshop on digital communications 2002 (invited paper)*, Capri, Italy, September 2002.

[WB97] Xiaolin Wu and P. Bao. Near-lossless image compression by combining wavelets and calic. In *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, volume 2, pages 1427–1431 vol.2, Nov 1997.

[WBSS04] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment : from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4) :600–612, April 2004.

[WC02] J.W. Woods and P.S. Chen. Improved mc-ezbc with quarter-pixel motion vectors. ISO/IEC/JTC1 SC29/WG11 doc no. m8366, Fairfax, May 2002.

[WDJ06] K. Wahid, V. Dimitrov, and G. Jullien. New encoding of 8x8 dct to make h.264 lossless. In *Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on*, pages 780–783, Dec. 2006.

[WLS07] Zhou Wang, Qiang Li, and Xinli Shang. Perceptual image coding based on a maximum of minimal structural similarity criterion. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 2 :II –121–II –124, 16 2007-Oct. 19 2007.

[WM97] Xiaolin Wu and Nasir Memom. Context-based, adaptative, lossless image coding. *IEEE TRANSACTIONS ON COMMUNICATIONS*, 45(4) :437–444, APRIL 1997.

[WQ05] Xiaolin Wu and Tong Qiu. Wavelet coding of volumetric medical images for high throughput and operability. *Medical Imaging, IEEE Transactions on*, 24(6) :719–727, June 2005.

[WSB03] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. *Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on*, 2 :1398–1402 Vol.2, Nov. 2003.

[WSS96] M. Weinberger, G. Seroussi, and G. Sapiro. LOCO-I : A Low Complexity, Context-Based, Lossless Image Compression Algorithm. In *Proc. IEEE Data Compression Conference*, Snowbird, Utah, March-April 1996.

[WSS00] M.J. Weinberger, G. Seroussi, and G. Sapiro. The loco-i lossless image compression algorithm : principles and standardization into jpeg-ls. *Image Processing, IEEE Transactions on*, 9(8) :1309–1324, Aug 2000.

[WT01] Yung-Gi Wu and Shen-Chuan Tai. Medical image compression by discrete cosine transform spectral similarity strategy. *Information Technology in Biomedicine, IEEE Transactions on*, 5(3) :236–243, Sept. 2001.

[WWJ+08] L. Wang, J. Wu, L.C. Jiao, L. Zhang, and G.M. Shi. Lossy to lossless image compression based on reversible integer dct. In *ICIP08*, pages 1037–1040, 2008.

[XWYP98] Zixiang Xiong, X. Wu, D.Y. Yun, and W.A. Pearlman. Progressive coding of medical volumetric data using three-dimensional integer wavelet packet transform. *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, pages 553–558, Dec 1998.

[YDD03] Hua Ye, Guang Deng, and John C. Devlin. A weighted least squares method for adaptive prediction in lossless image compression. In *Picture Coding Symposium*, pages 489–493, Saint-Malo, France, 2003.

[ZL77] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *Information Theory, IEEE Transactions on*, 23(3) :337–343, May 1977.

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399