Télédétection, 2006, vol. 6, n° 4, p. 345-360
© 2006 CONTEMPORARY PUBLISHING INTERNATIONAL
Publié sous l'enseigne *Éditions Scientifiques GB*

ALGORITHME NON ITÉRATIF BASÉ SUR LES ONDELETTES BIORTHOGONALES ET LES FRACTALES POUR LA COMPRESSION D'IMAGES SATELLITAIRES

Mourad LAHDIR [1], Soltane AMEUR [2*], et Abd El Hamid ADANE [3]

[1] *Université Mouloud Mammeri de Tizi-Ouzou (U.M.M.T.O), Faculté de Génie Électrique et Informatique, Département d'Électronique, Laboratoire d'Analyse et de Modélisation des Phénomènes Aléatoires (L.A.M.P.A), B.P. n°17 RP 15000, Tizi-Ouzou, Algérie.*
Courriel : mlahdir@yahoo.fr
[2] *Université Mouloud Mammeri de Tizi-Ouzou (U.M.M.T.O), Faculté de Génie Électrique et Informatique, Département d'Électronique, Laboratoire d'Analyse et de Modélisation des Phénomènes Aléatoires (L.A.M.P.A), B.P. n°17 RP 15000, Tizi-Ouzou, Algérie.*
Téléphone / Télécopieur : 00 213 26 21 86 42; courriel : ameursoltane@yahoo.com
[3] *Université des Sciences et de la Technologie d'Alger (U.S.T.H.B.), Faculté d'Électronique et d'Informatique, Département Télécommunications, Laboratoire Traitement d'Images et Rayonnement (L.T.I.R), BP n° 32 El Alia, Bab Ezzouar, 16111, Alger, Algérie.*
Courriel : aadane@usthb.dz

Soumis le 11 décembre 2006 ; révisé le 28 février 2007 ; accepté le 26 mars 2007

Un algorithme rapide basé sur les ondelettes biorthogonales et les fractales a été mis au point pour faciliter l'archivage d'images de grande taille avec de forts taux de compression et le minimum de distorsion. Pour cela, l'image à traiter est compressée puis décompressée en utilisant des ondelettes biorthogonales et une quantification vectorielle. L'image d'ondelettes qui en résulte est ensuite soustraite de l'image originale pour donner une image résiduelle. Cette dernière est codée par des fractales en créant un domaine de contraction avec les blocs « source » extraits de l'image d'ondelettes et un domaine de destination obtenu à partir de l'image résiduelle. Les paramètres du code fractal sont ensuite déterminés en collant les blocs « destination » aux blocs « source ». Ces paramètres et ceux de la quantification vectorielle sont codés en utilisant l'algorithme de Huffman pour générer la séquence binaire représentant l'image compressée. Des transformations inverses permettent de décompresser l'image. À la différence des systèmes de fonctions itérées locales (LIFS), notre méthode n'exige aucune itération car l'opération de collage est réalisée en une seule étape. Les tests sur une image du visage d'une femme donnent un taux de compression atteignant 94 % et un PSNR de 32,9 dB. De plus, la durée du traitement est près de 200 fois plus courte qu'avec les LIFS. Cette méthode, testée sur diverses images satellitaires, a été employée pour archiver des images du canal à haute résolution dans le visible de METEOSAT acquises à toutes les 15 minutes en 2003. Après décompression, ces images sont reproduites fidèlement et l'espace-mémoire requis pour leur stockage est réduit de plus de 90 %.

Mots-clés: Télédétection ; compression ; ondelettes ; quantification ; fractales.

NON ITERATIVE ALGORITHM BASED ON BIORTHOGONAL WAVELETS AND FRACTALS FOR THE COMPRESSION OF SATELLITE IMAGES

A fast algorithm based on both biorthogonal wavelets and fractals is implemented to facilitate the storage of large format images with high compression ratios and minimum loss of information. For this, the image to be processed is compressed and decompressed using biorthogonal wavelets at different resolutions and a vector quantization technique. An image of wavelets is then obtained and removed from the original ones. The resulting residual image is coded with fractals by creating a contraction domain using the source blocks extracted from the wavelet image and a destination domain deduced from the residual image. The fractal coding parameters are determined by pasting the destination to source blocks. To generate the binary sequence representing the compressed image, the fractal and vector quantization parameters are coded using the Huffman algorithm. Inverse transformations enable us to decompress the image. Unlike the local iterated function system (LIFS), the above method requires no iteration since the pasting operation is achieved following only one step. The tests performed on an image of a woman's face yield a compression ratio reaching 94 % and a PSNR of 32,9 dB. In addition, processing time is about 200 times shorter than for LIFS. This method was tested on various satellite images and then used to store high resolution visible METEOSAT images (MSG) collected every 15 minutes during 2003. After decompression, these images are faithfully reproduced and the memory space needed for their storage is reduced by more than 90 %.

Keywords: Remote sensing ; compression ; wavelets ; quantization ; fractals.

1. INTRODUCTION

Avec le développement de l'informatique, des appareils très performants sont apparus sur le marché et de puissantes méthodes ont été mises en œuvre pour faire face aux besoins d'acquisition, de stockage et de traitement de l'information (Marsault, 1995). Un tel développement a engendré des applications qui mettent en jeu la transmission et l'archivage de données de plus en plus

volumineuses. Il est alors devenu primordial de compresser ces données pour les ramener à une taille plus facilement exploitable. De façon générale, les fichiers numériques à traiter se distinguent par leur format, qui est identifiable grâce à l'extension du nom du fichier. Dans le cas d'images brutes, le fichier bitmap reste le plus utilisé. En fait, près de 70 formats différents sont actuellement disponibles pour stocker des images bitmap, tout en étant conformes aux standards les plus usités, tel que Windows ou Unix. Ceci implique la compatibilité des programmes avec de telles normes, lors de leur conception. Si l'on veut augmenter la vitesse de consultation des fichiers ou minimiser l'espace de stockage des documents numérisés, surtout dans le cas des images, il est nécessaire de recourir à des systèmes de compression appropriés. Deux sortes de techniques permettent la compression des images : les méthodes réversibles, c'est-à-dire sans pertes, qui conduisent à de faibles taux de compression et celles appelées irréversibles qui permettent de compresser fortement les images, mais au prix de certaines distorsions (Marsault, 1995 ; Held and Marshall, 1996).

Parmi les méthodes dites sans pertes, citons la technique RLC (*Run Length Coding*), le codage de Huffman et la compression LZW (*Lempel Ziv Welch*). Ces méthodes opèrent avec des facteurs de compression allant de 1,2 à 2,5.

Quant aux méthodes à fort taux de compression, les plus efficaces sont basées sur de puissants outils, tels que la transformée discrète en cosinus (DCT), la transformée en ondelettes ou l'approche fractale (Jacquin, 1992 ; Fisher, 1994 ; Marsault, 1995 ; Averbuch *et al.*, 1996 ; Wohlberg *et al.*, 1999 ; Ghosh *et al.*, 2004). Avec de tels outils, le fichier image subit une forte compression, mais avec des pertes se traduisant par des dégradations plus ou moins perceptibles à l'œil nu lors de la décompression. Parmi les méthodes à fort taux de compression, on distingue le format JPEG (*Joint Photographic Expert Group*) (Borko, 1995) et la méthode GIF (*Graphic Interchange Format*) (Held and Marshall, 1996). JPEG est actuellement le plus couramment employé. Son principe est de diviser l'image en blocs carrés de 8 x 8 pixels et de coder les valeurs les plus proches dans chaque bloc sur quelques bits. En procédant de la sorte, on peut atteindre des taux de compression supérieurs à 90 %, tout en restituant des images avec une perte d'informations à peine perceptible. Mais, à l'instar d'autres méthodes irréversibles, le temps d'exécution est très lent, surtout dans le cas d'images de grande taille. De plus, des artefacts apparaissent dans celles-ci sous forme de mosaïques. Pour réduire les distorsions causées par les méthodes à fort taux de compression, nous avons élaboré, lors de travaux précédents, une méthode de compression basée sur la DCT et la quantification vectorielle.

Cette technique appliquée à des images de référence, comme l'image de Lena ou les images de Brodatz et à des images de télédétection prises par le capteur de METEOSAT-4, a donné des résultats satisfaisants (Ameur *et al.*, 2002). En effet, elle a permis de réaliser un bon compromis entre le taux de compression et la qualité de restitution des images. Grâce à cette méthode, le taux de compression et le PSNR atteignent respectivement 94 % et 33 dB.

Cependant, l'apparition de nouvelles technologies de communication impose la manipulation d'un volume de données de plus en plus important qui doit être traité dans des délais de plus en plus courts, avec le meilleur compromis possible entre le taux de compression et la qualité de restitution des images. Certaines techniques de compression, basées sur les fractales, semblent convenir à cette exigence (Bedford *et al.*, 1992 ; Zhao and Yuan, 1994 ; Davoine, 1995 ; Davis, 1998 ; Lepsy *et al*, 1998 ; Hamzaoui and Saupe, 2000 ; Belloulata, 2005). En effet, ces méthodes partent du principe que toute image peut être considérée comme résultant de la répétition de mêmes motifs à différentes échelles. Pour mettre en œuvre ces techniques, un système de fonctions itérées locales (ou LIFS, i.e., *Local Iterated Function System*) (Jacobs *et al.*, 1992 ; Jacquin, 1992 ; Fisher, 1994), formé de fonctions contractantes est défini. Le théorème de collage mettant en jeu ces fonctions permet alors de représenter tout objet à deux dimensions sous forme de fractales.

De telles techniques ont l'intérêt d'accélérer le processus de compression tout en restituant des images de bonne qualité. Aussi, nous avons mis à profit les avantages des ondelettes biorthogonales, de la quantification vectorielle et des fractales pour élaborer un algorithme de compression d'images opérant sans aucune itération. Cet algorithme et les résultats obtenus sont présentés dans les prochaines sections.

2. MÉTHODOLOGIE

2.1. Principe

Considérons une image f(i, j) à compresser, dans laquelle la position d'un pixel est repérée par ses coordonnées (i, j). Le schéma de la figure 1 représente la structure de l'algorithme que nous avons conçu, élaboré et appliqué à diverses images. Comme le montre ce schéma, une transformation en ondelettes biorthogonales, assortie d'une quantification vectorielle, est d'abord appliquée à l'image f(i, j). Ce type de quantification est couramment employé en compression d'images (Moureau, 1994 ; Averbuch *et al.*, 1996 ; Cosman *et al.*, 1996). C'est ainsi que l'association de la quantification vectorielle aux transformées en ondelettes et en cosinus nous a déjà permis de traiter efficacement des images des capteurs de METEOSAT (Ameur *et al.*, 2002).

Grâce au processus de compression/décompression décrit sur la figure 1, une image F(i, j), compor-

tant le maximum d'informations, est obtenue. En effectuant la différence des deux images f(i, j) et F(i, j), on obtient une image « résidu » r(i, j). Le codage par des fractales est ensuite appliqué à ce « résidu » pour prendre en compte les détails de l'image originale f(i, j**)** qui n'ont pas pu être inclus dans l'image F(i, j). À cet effet, r(i, j) est partagé en blocs « destination », constituant alors le domaine de destination. Parallèlement, pour accélérer le processus de codage, F(i, j) subit une décimation par quatre pour donner une image de taille plus réduite d(k, l). Cette dernière est composée de blocs « source », formant ainsi le domaine de contraction. Le codage par des fractales revient à transformer les blocs « source » de façon qu'ils puissent approximer au mieux les blocs « destination ». Le code fractal ainsi défini est exprimé en fonction de la position p du bloc « source », d'un facteur d'échelle α et d'un paramètre d'isométrie I, caractérisant les fractales de l'image étudiée. En pratique, pour obtenir ce code nous avons collé les blocs « source » aux blocs « destination » de l'image r(i, j) en une seule étape. Finalement, ces paramètres et ceux provenant de la quantification vectorielle sont codés simultanément à l'aide de l'algorithme de Huffman pour générer une séquence binaire contenant l'information utile à sauvegarder, c'est-à-dire l'image compressée. La décompression des images est alors réalisée en procédant par transformations inverses.

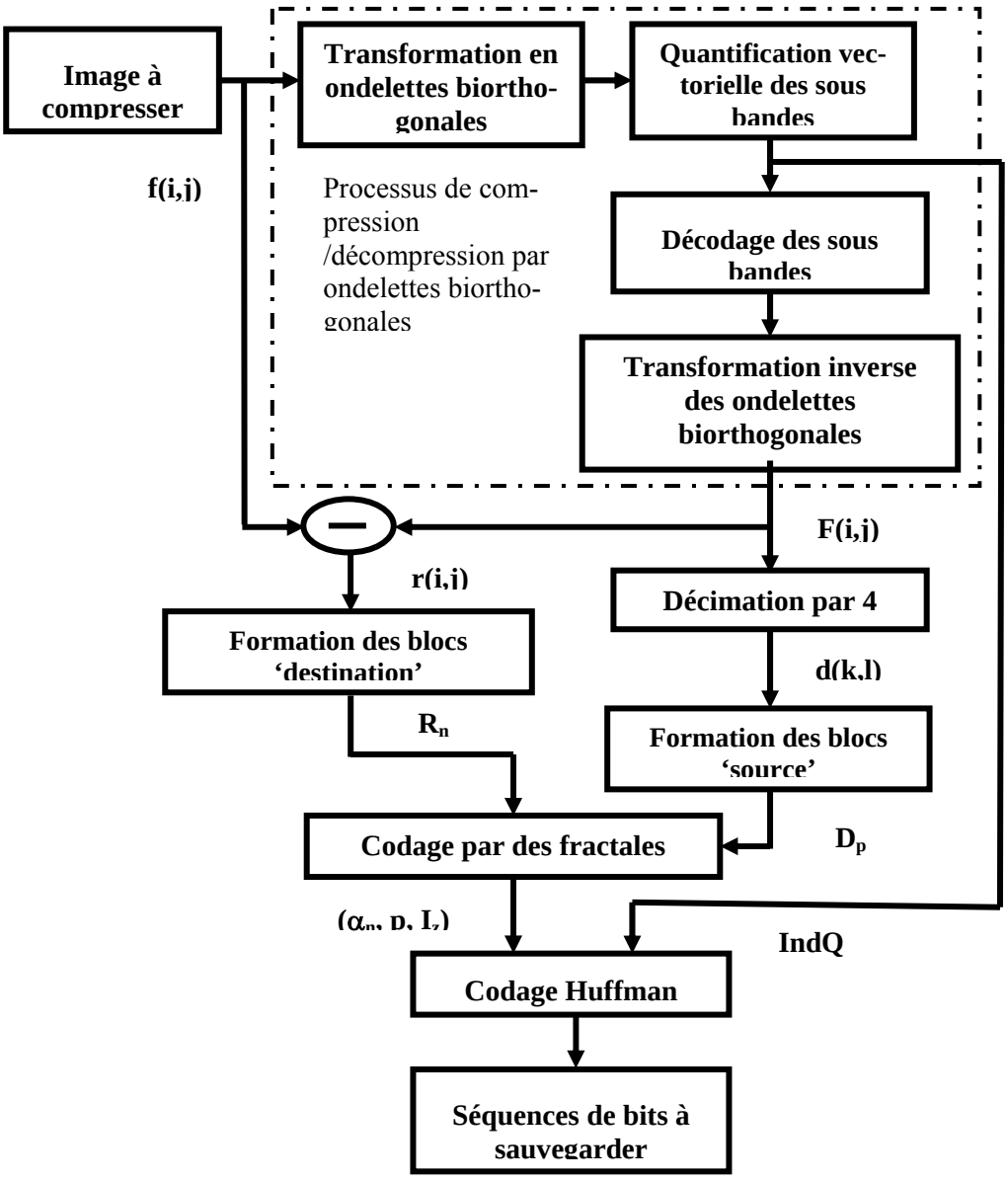


Figure 1 : Schéma du processus de compression d'images. *Scheme of the image compression process.*

2.2. Décomposition en ondelettes biorthogonales

Pour former le domaine de contraction du codage, l'image à compresser f(i, j) est d'abord décomposée en ondelettes à différents niveaux de résolution, selon une hiérarchie pyramidale. Pour ce faire, l'algorithme de Mallat (1989) est appliqué à l'image à traiter. Une telle décomposition est établie en partant de la formulation générale qui permet de définir une transformée en ondelettes biorthogonales à une seule dimension x. En effet, cette transformée est mise en œuvre en considérant deux ondelettes mères ψ et $\tilde{\psi}$. Ces deux sortes d'ondelettes sont, à la fois, dilatées et translatées de façon à générer deux familles d'ondelettes $\psi_{m,n}$ et $\tilde{\psi}_{m,n}$, où m et n jouent respectivement le rôle d'indice de dilatation et d'indice de translation. Pour simplifier l'écriture, posons : $\Omega = \psi$ ou $\tilde{\psi}$. Chacune de ces familles d'ondelettes est alors de la forme :

$$\Omega_{m,n}(x) = 2^{-m/2}\, \Omega(2^{-m}\, x - n) \qquad (1)$$

avec m = 1 ; 2 …… et n = 0 ; 1 ; 2

En général, les ondelettes $\psi_{m,n}$ et $\tilde{\psi}_{m,n}$ constituent une base duale non orthogonale dans l'espace des fonctions de carrés sommables $L^2(\Re)$, défini dans le domaine des réels $(\Re)$ (Antonini *et al.*, 1992). Par suite, la transformée en ondelettes biorthogonales consiste à représenter toute fonction f(x) appartenant à l'espace $L^2(\Re)$ par la combinaison linéaire suivante :

$$f = \sum c_{m,n}(f)\, \tilde{\psi}_{m,n} \qquad (2)$$

Les coefficients d'ondelettes $c_{m,n}(f)$ sont ensuite calculés, en effectuant le produit scalaire de la fonction f avec les ondelettes $\psi_{m,n}$, soit :

$$c_{m,n}(f) = \langle \psi_{m,n}\, , f \rangle = \int \psi_{m,n}(x)\, f(x)\, dx \qquad (3)$$

Pour représenter la fonction f(x) à différentes résolutions, deux fonctions d'échelle ϕ et $\tilde{\phi}$ ainsi que deux familles d'ondelettes $\phi_{m,n}$ et $\tilde{\phi}_{m,n}$ sont considérés. En prenant $\Omega = \phi$ ou $\tilde{\phi}$, on en arrive à :

$$\Omega_{m,n}(x) = 2^{-m/2}\, \Omega(2^{-m}\, x - n) \qquad (4)$$

avec m = 1 ; 2 …… et n = 0 ; 1 ; 2

Soit, V_m l'espace fermé engendré par la famille d'ondelettes $\tilde{\phi}_{m,n}$ pour *m* fixé. La projection de f sur l'espace V_m donne une bonne approximation de cette fonction à la résolution 2^{-m}, en écrivant que :

$$\mathrm{Proj}(f / V_m) = \sum \langle \varphi_{m,n}\, , f \rangle\, \tilde{\varphi}_{m,n} \qquad (5)$$

Puis, l'approximation de l'image à l'échelle 2^{-m+1} est obtenue en faisant :

$$\mathrm{Proj}(f / V_{m-1}) = \mathrm{Proj}(f\, , V_m) + \sum c_{m,n}(f)\, \tilde{\psi}_{m,n} \qquad (6)$$

Dans cette expression, les coefficients d'ondelettes $c_{m,n}$ traduisent l'information perdue entre la représentation de la fonction f à la résolution 2^{-m} et celle à la résolution 2^{-m+1}.

2.3. Décomposition en sous bandes

La transformée en ondelettes biorthogonales décrite précédemment a été appliquée à l'image f(i, j), en calculant les projections définies par les équations (5) et (6) par filtrage. Pour ce faire, nous avons utilisé deux paires de filtres : une paire $(h\, , g)$ pour la décomposition et une autre $(\tilde{h}\, , \tilde{g})$ pour la reconstruction. Par définition, $h(u)$ et $\tilde{h}(u)$ sont les réponses impulsionnelles de deux filtres passe-bas associés aux fonctions ϕ et $\tilde{\phi}$. De même $g(u)$ et $\tilde{g}(u)$ sont les réponses impulsionnelles de deux filtres passe-haut associés aux fonctions ψ et $\tilde{\psi}$.

L'application de ces filtres à l'image à traiter permet d'extraire les coefficients de basses fréquences A_J et les coefficients d'ondelettes C_J. Les coefficients A_J représentent l'approximation de l'image au niveau de résolution ultime J et les coefficients C_J, les détails perdus entre deux niveaux de résolution successifs, dans les directions horizontale, verticale et diagonale. Ils s'écrivent de la façon suivante :

$$A_J = \sum_u A_{J-1}(2n - u)\, h(u) \qquad (7)$$

$$C_J = \sum_u A_{J-1}(2n - u)\, g(u) \qquad (8)$$

avec u = 1 ; 2 ….

Les types de filtres biorthogonaux et la plus grande résolution J avec lesquels nous avons travaillé, ont été choisis après différents essais. Nous avons alors trouvé que, pour reproduire toute image avec la meilleure qualité possible, il fallait fixer la résolution ultime à 2^{-3} et utiliser des filtres biorthogonaux 9/7 (Antonini *et al.*, 1992 ; Daubechies *et al.*, 1992). Sur la figure 2, on sché-

matise le processus de décomposition d'une image quelconque en sous bandes selon les résolutions 2^{-1}, 2^{-2} et 2^{-3}. Sur cette figure, les sous bandes de détails caractérisant les coefficients d'ondelettes aux résolutions 1, 2 et 3 sont désignées par les blocs C1, C2 et C3. La sous bande lissée à la résolution 3 est représentée par le bloc A3. Les détails horizontaux sont repérés par la lettre (h), les détails verticaux par la lettre (v) et les détails diagonaux par la lettre (d).

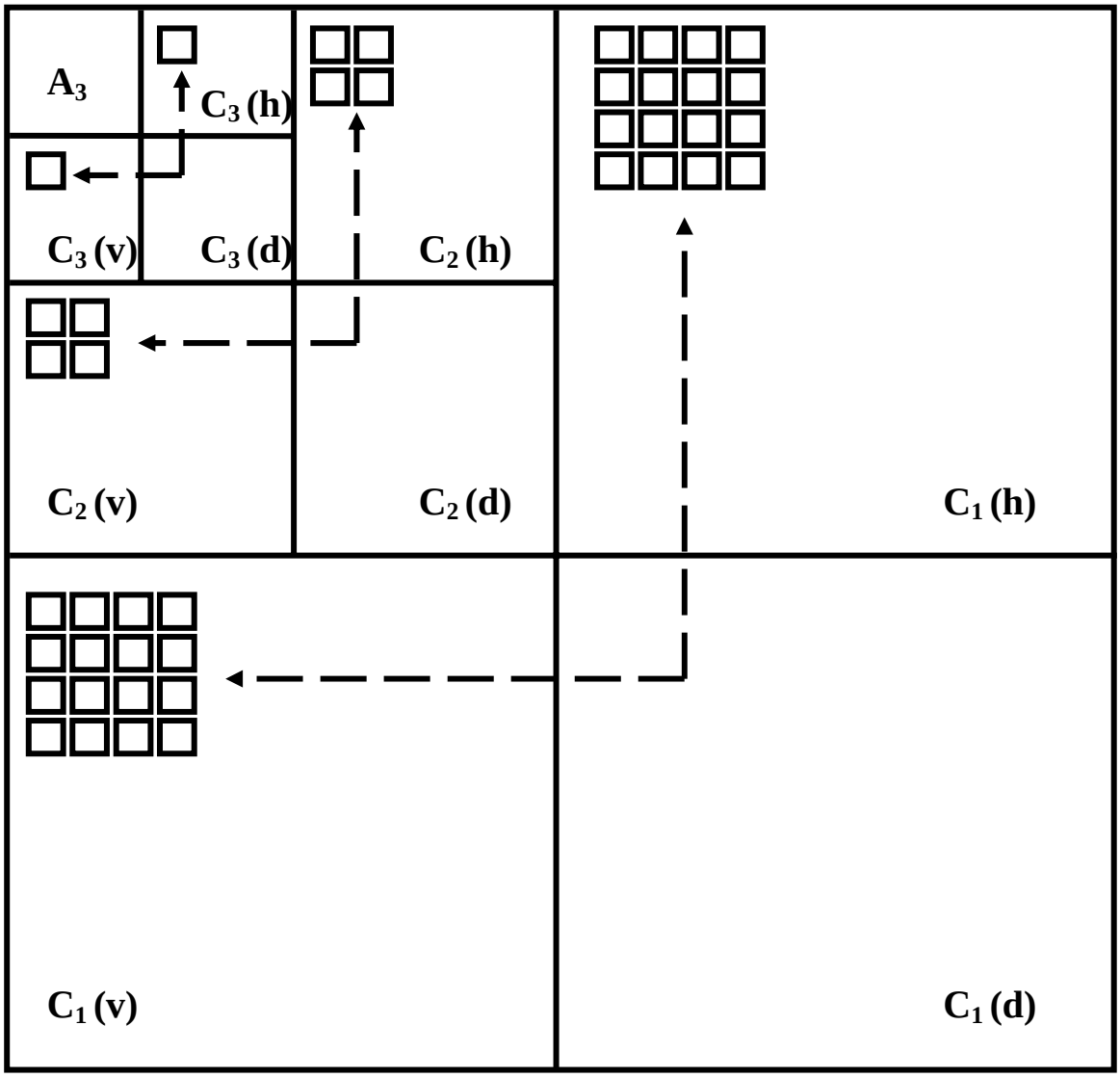


Figure 2 : Décomposition d'une image en sous bandes et quantification vectorielle. *Image decomposition into subbands and vector quantization.*

2.4. Quantification vectorielle des sous bandes

Après l'étape de décomposition en ondelettes biorthogonales, les sous bandes de détails sont codées par quantification vectorielle. Cette dernière consiste à établir un dictionnaire formé d'un ensemble de vecteurs qui soit le plus près possible de la partition initiale. Pour confectionner ce dictionnaire, nous avons utilisé l'algorithme d'apprentissage par compétition (ou CL, i.e., *Competitive Learning*). La structure de cet algorithme est décrite dans l'une de nos études (Ameur *et al.*, 2002). Comme l'indique la figure 2, les coefficients d'ondelettes obtenus dans les directions horizontale, verticale et diagonale ont été évalués à chaque résolution à l'aide d'un seul dictionnaire pour déterminer les indices de quantification (Ind Q) à sauvegarder.

2.5. Reconstitution, décimation et détermination du résidu

L'application des transformations inverses au processus de compression par ondelettes biorthogonales permet de reconstituer l'image F(i, j). Cette image est d'abord partagée en blocs B(i, j), ayant chacun une taille de 4 x 4 pixels**.** Puis, comme nous l'avons expliqué à la section 2.1, cette image subit une décimation par quatre pour former une image de taille plus réduite d(k, l). Dans cette image, la position d'un pixel est repérée par ses coordonnées (k, l) et, à chaque pixel de l'image, est attribuée la valeur moyenne des pixels du bloc B(i, j) correspondant. L'image « résidu » r(i, j) est obtenue en faisant la différence, pixel par pixel, des deux images f(i, j) et F(i, j). Ce dernier représente la perte d'information dans le processus de compression/décompression par ondelettes biorthogonales et par quantification vectorielle. Par la suite, l'image réduite d(k, l) et le « résidu » r(i, j) seront utilisés pour générer respectivement les domaines de contraction et de destination.

2.6. Codage par des fractales

Pour prendre en compte les détails de l'image originale f(i, j) non encore traités, nous avons codé l'image « résidu » r(i, j) par des fractales. Ce choix est motivé par le fait que la structure d'une image est formée d'un ensemble de motifs semblables, mais ayant des tailles et des orientations différentes (Marsault, 1995). Aussi, nous pouvons faire l'hypothèse que le « résidu » se compose de fractales qui donnent une information supplémentaire sur la structure de l'image f(i, j). Pour identifier ces motifs, l'image « résidu » r(i, j) subit une partition en n blocs de destination R_n, ayant chacun une taille de 4 x 4 pixels et où n est le numéro du bloc considéré dans le domaine de destination. Parallèlement, le domaine de contraction est obtenu en décomposant l'image réduite d(k, l) en p blocs « source » D_p de même taille que les blocs « destination » et où p est le numéro du bloc considéré dans ce domaine. Le codage par des fractales consiste à déterminer une transformée affine et contractante w_n qui, en l'appliquant aux blocs « source » D_p, donne la meilleure approximation possible des blocs « destination » R_n. À la suite de cette transformation, les paramètres caractéristiques des fractales qui permettent d'obtenir des blocs similaires sont sauvegardés (figure 3).

En pratique, la transformation w_n que nous avons utilisée permet de modifier la luminance et la forme des blocs « source » (Jacquin, 1992 ; Fisher, 1994). Une telle opération est réalisée en faisant varier, simultanément, le facteur d'échelle α et le facteur d'isométrie I décrivant les fractales. Le premier paramètre agit sur le contraste et le second sur la déformation géométrique. Cette dernière est obtenue en mélangeant les pixels et en les déplaçant à l'intérieur des blocs. D'où :

$$w_n\left(D_p\right) = \alpha_n \, I_z\left(D_p\right) \tag{9}$$

Dans cette expression la valeur de α est comprise entre 0 et 1, tandis que I peut prendre huit valeurs différentes en fonction de l'orientation considérée (tableau 1). Pour identifier un bloc « destination » R_n, il suffit de le comparer à l'ensemble des blocs « source » D_p. Pour trouver le bloc D_p le plus proche du bloc R_n, la distance euclidienne séparant les blocs D_p et R_n, est minimisée, en faisant varier le facteur d'échelle α et le facteur d'isométrie I. Cette distance s'écrit :

$$E_{pn} = \sum_{i=0}^{i=3} \sum_{j=0}^{j=3} \left(R_n(i,j) - \alpha_n \, I_z\left\{D_p(i,j)\right\}\right)^2 \tag{10}$$

où α_n est le facteur d'échelle du bloc « source » de position p et I_z représente l'une des huit isométries utilisées, avec $I_z = I_1 ; I_2 ; \ldots, I_8$.

Cette opération est ensuite répétée pour tous les blocs D_p. Soit (α_n, p, I_z), les paramètres du codage fractal qui minimisent la distance E_{pn}. L'étape finale est la production de la séquence binaire caractérisant l'image compressée. À cet effet, les paramètres (α_n, p, I_z) et les indices de quantification vectorielle (Ind Q) sont codés simultanément à l'aide de l'algorithme de Huffman.

2.7. Décompression

Le schéma de décompression que nous avons adopté est présenté sur la figure 4. Dans ce schéma, la séquence binaire provenant du processus de compression est également décodée à l'aide de l'algorithme de Huffman. Cette opération permet de restituer les indices de quantification vectorielle (Ind Q) et les paramètres (α_n, p, I_z) issus du codage par des fractales. Les indices Ind Q sont utilisés pour reconstruire les différentes sous bandes A3, C1, C2 et C3. Puis, l'image F(i, j) est reconstituée en appliquant une transformée inverse en ondelettes biorthogonales à ces quatre sous bandes. Comme on l'a fait précédemment, cette image subit une décimation par quatre pour former l'ensemble des blocs « source » D_p constituant l'image d(k, l). Les blocs « destination » R_n de l'image « résidu » r(i, j) à reconstituer sont obtenus par décodage des blocs « sources » D_p. Pour ce faire, la transformation suivante est appliquée :

$$R_n\left(i,j\right) = \alpha_n \, I_z\left\{D_p\left(i,j\right)\right\} \tag{11}$$

L'image décompressée F'(i, j) est finalement obtenue en additionnant les images F(i, j) et r(i, j), pixel par pixel.

3. DONNÉES UTILISÉES

Pour élaborer l'algorithme décrit dans la section précédente et évaluer ses performances, deux sortes d'images ont été utilisées:

l'image de Lena, qui est un portrait de référence composée de diverses textures, dont des textures

très fines comme les yeux et les cheveux (figure 5) ;

une série d'images d'un capteur de METEOSAT acquises toutes les 15 minutes dans le canal du visible pendant l'année 2003 ; ces images ont été prises par un capteur du satellite METEOSAT de seconde génération (MSG), qui est un satellite

géostationnaire doté de douze canaux différents du spectre dans les bandes du visible et l'infrarouge ; à titre indicatif, les principales caractéristiques de ces canaux sont présentées sur le tableau 2.

Sur la figure 6, on présente un exemple d'image du capteur de METEOSAT que nous avons traitée. Cette image est formée de 512 x 512 pixels, codés sur 256 niveaux de gris. Elle a été acquise dans le canal à haute résolution visible (HRV au-dessus de l'Afrique du Nord, le 11 novembre 2003 à 12 h GMT.

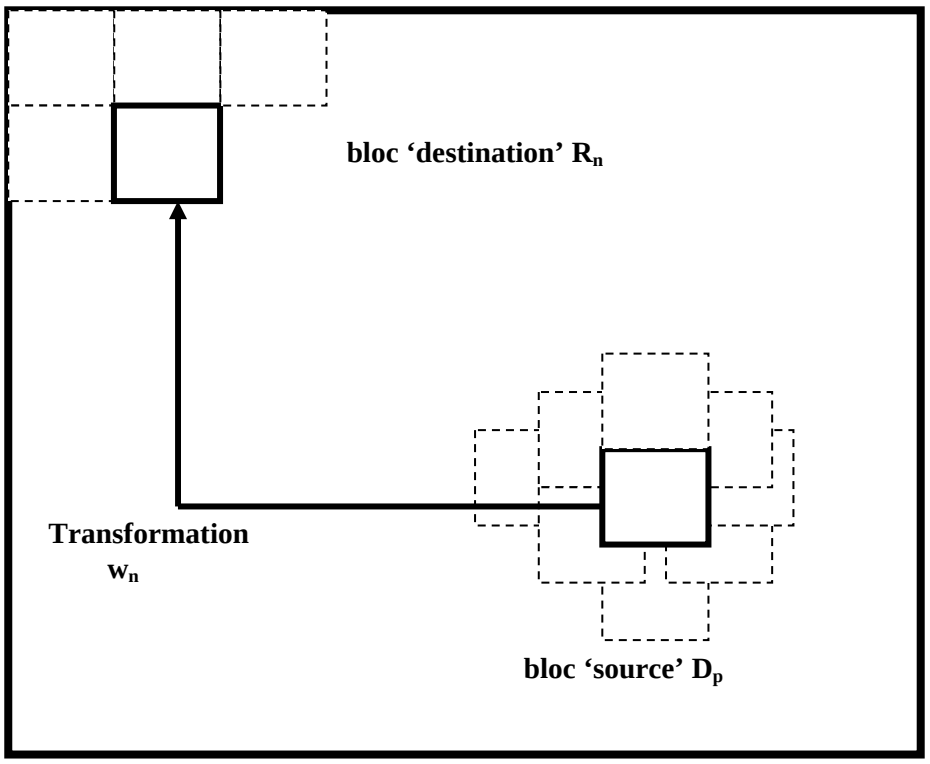


Figure 3 : Collage d'un bloc « source » sur un bloc « destination ». *Pasting of "source" towards "destination" block.*

4. RÉSULTATS ET DISCUSSION

Pour évaluer ses performances, l'algorithme de compression/décompression représenté par les organigrammes des figures 1 et 4, a été appliqué, dans un premier temps, à l'image de Lena. Pour ce faire, nous avons utilisé un microordinateur Pentium 4 dont la fréquence de travail est de 2,4 GHz et qui est doté d'une mémoire centrale de 256 Mo et d'un disque auxiliaire de 40 Go. Comme le montre la figure 7, l'image restituée par cet algorithme est de bonne qualité, mais accompagnée d'artefacts très minimes. Pour mieux apprécier la pertinence de ces résultats, deux grandeurs caractéristiques ont été considérées. Ce sont le taux de compression (C) et le rapport signal sur bruit (PSNR). Ce dernier étant exprimé en valeurs de crête, est déterminé en comparant l'image reconstruite après compression/décompression, à l'image originale. Ces deux grandeurs sont, par définition :

$$C = 100\left(1 - \frac{bpp}{8}\right) \quad (12)$$

$$PSNR = 10\log\left(\frac{255^2}{MSE}\right) \quad (13)$$

Dans l'équation (12), bpp est le nombre de bits par pixel après codage et, dans l'équation (13), MSE représente l'erreur moyenne quadratique, calculée en considérant les écarts existant entre l'image originale et l'image reconstruite.

La courbe de la figure 8 illustre les variations du rapport signal sur bruit en fonction du facteur de compression pour l'image de Lena. On voit que,

pour de forts taux de compression, le rapport signal sur bruit avoisine 30 dB, prouvant ainsi que l'image restituée est de bonne qualité. En particulier, pour un taux de compression de 93,75 %, on trouve que le PSNR est de 32,9 dB. Un tel résultat montre tout l'intérêt de tenir compte de l'information contenue dans l'image « résidu » lors du processus de compression/décompression.

Dans une seconde étape, les résultats obtenus avec l'image de Lena ont été comparés à ceux de diverses méthodes publiées dans la littérature (Jacquin, 1993 ; Lepsy *et al.*, 1993 ; Thomas and Deravi, 1995 ; Davis, 1998 ; Chung-Su, 1998 ; Charier *et al*., 1999 ; Ameur *et al.*, 2002). De telles méthodes sont habituellement basées sur les systèmes de fonctions itérées locales (LIFS), sur les ondelettes ou sur la DCT. Celles que nous avons choisies de comparer sont répertoriées sur le tableau 3.

Le but de la comparaison est d'identifier les méthodes qui conduisent à un taux de compression et un PSNR les plus grands possibles, avec un temps d'exécution le plus court possible. Pour faciliter cette comparaison, nous avons défini un facteur de performance (Q), en procédant par analogie avec le facteur de mérite d'un amplificateur. Le facteur (Q) est alors obtenu en effectuant le produit du taux de compression et du PSNR convertis en rapports sans dimensions.

D'après le tableau 3, la méthode FVQ (*Fractal Vector Quantizer*), puis notre algorithme, sont les plus performants du point de vue de la compression et de la restitution d'une image. Les temps de compression et de décompression ont été estimés en fonction du type d'approche et du nombre d'opérations à mener. Cette estimation a été effectuée en traitant l'image de Lena à l'aide de programmes spécifiques. Comme on l'indique sur le tableau 4, le temps de décompression est pratiquement le même, quelle que soit l'approche employée. Pour le processus de compression, notre algorithme opère avec un temps beaucoup plus court que les méthodes basées sur les LIFS, car ces dernières sont presque 200 fois plus lentes.

La rapidité de cet algorithme s'explique par la réduction du nombre de blocs « source » pris en considération lors de l'opération de collage par des fractales et par le codage de ces blocs, qui est effectué sans aucune itération. En effet, le temps nécessaire à l'opération de collage est proportionnel au nombre de pixels constituant les blocs de codage (Marsault, 1995).

En résumé, les résultats présentés sur les tableaux 3 et 4 montrent que l'algorithme décrit sur les figures 1 et 4 est celui qui réalise le meilleur compromis entre le taux de compression, la qualité de restitution de l'image et la rapidité d'exécution des calculs.

Dans une dernière étape, cet algorithme a été appliqué à l'image de la figure 6. Sur la figure 9, on présente l'image restituée après compression et décompression. Sur cette figure, les nuages, les catégories d'occupation des terres, la mer Méditerranée et l'océan Atlantique apparaissant sur l'image originale sont bien reproduits et les contours bien définis. La courbe de la figure 10 montre que le rapport signal sur bruit croît pratiquement comme le logarithme du facteur bpp. Donc, le PSNR décroît lentement avec le taux de compression (C) dans l'échelle des forts taux de compression. Notons que l'espace-mémoire que doit occuper la série d'images du capteur de METEOSAT, prises pendant toute l'année 2003, est de 110 Go avant compression. L'application de notre algorithme à toutes ces images a permis de ramener cet espace à 6 Go, soit un gain de 104 Go, rendant ainsi possible leur stockage dans notre microordinateur. Par la suite, cette bibliothèque d'images satellitaires ainsi sauvegardée pourra contribuer à l'élaboration d'un historique de l'évolution actuelle du climat.

Pour apprécier la flexibilité de notre algorithme, ce dernier a aussi été testé sur deux images de télédétection prises respectivement au-dessus de Montréal et de Washington. La première est une image de HRV de SPOT de taille 512 x 512 pixels, ayant une résolution de 10 m et la seconde est une image d'IKONOS formée de 706 x 401 pixels avec une résolution de 1 m. Pour C = 92,6 %, on trouve un PSNR de 23,17 dB et un temps d'exécution de 9 s pour l'image HRV. Avec ce taux de compression, le PSNR est de18,09 dB et le temps d'exécution de 12 s pour l'image d'IKONOS. Comme pour l'image du capteur de METEOSAT, on constate que le facteur de qualité est très élevé et le temps d'exécution, très court. Remarquons cependant qu'on ne peut pas traiter des images dont la taille devient supérieure à 1 000 x 1 000 pixels à cause de l'insuffisance des ressources de la mémoire du microordinateur Pentium 4 utilisé.

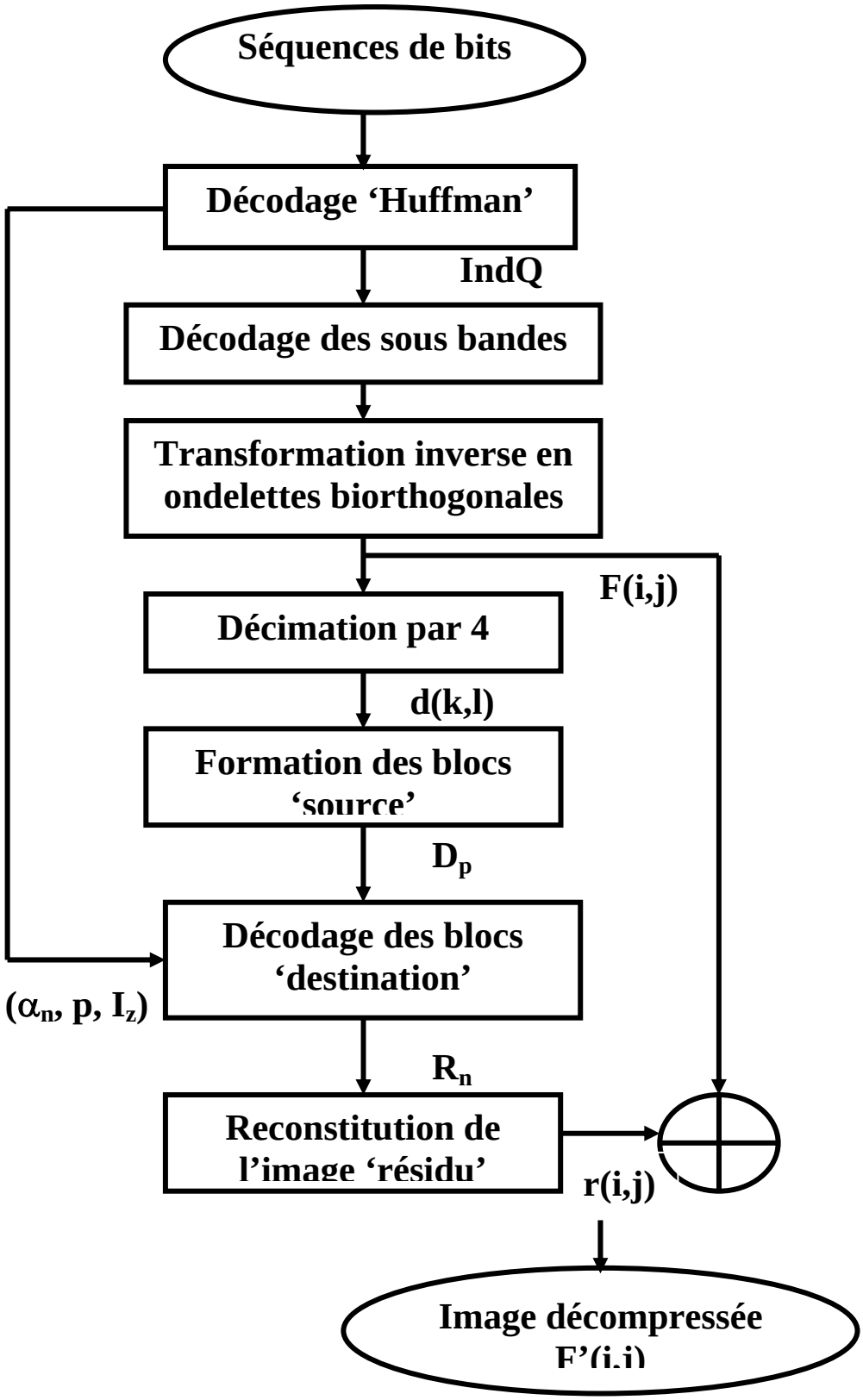


Figure 4 : Organigramme de décompression d'image. *Flowchart of image decompression.*



Figure 5 : Image de Lena. *Image of Lena.* (Echelle : 512*512 pixels)

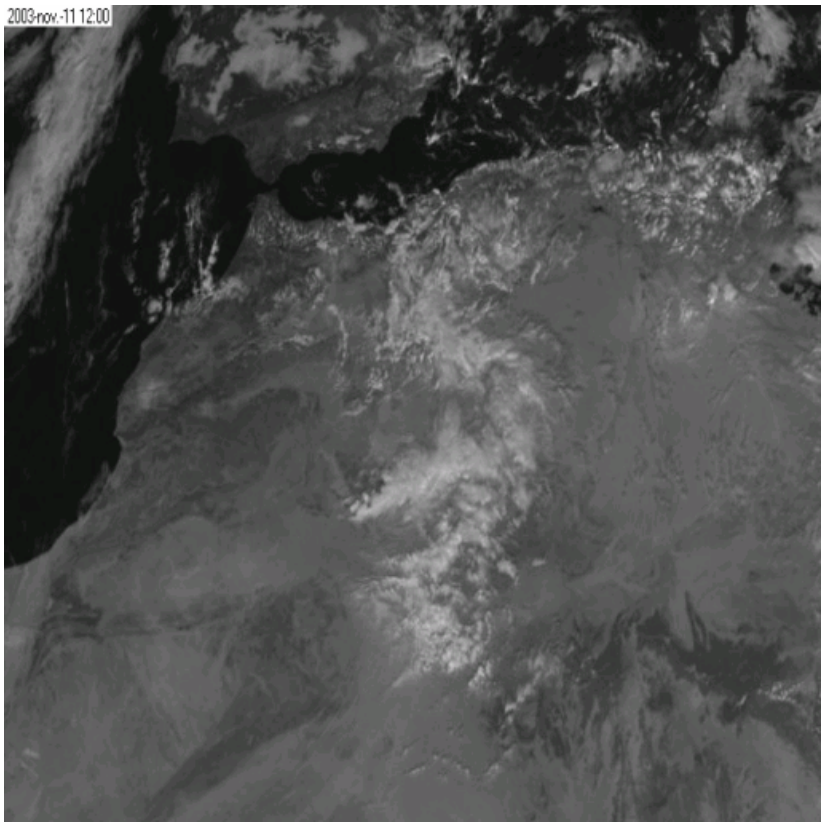


Figure 6 : Image du capteur de METEOSAT prise le 10 novembre 2003 à 12 h GMT. *Image of METEOSAT sensor acquired on November 10, 2003 at 12 h UTC.* (Echelle : 512*512 pixels)



Figure 7 : Reconstitution de l'image de Lena (bpp = 0,5 bits/pixel et PSNR = 32,9 dB). *Reconstitution of the image of Lena (bpp = 0,5 bits/pixel and PSNR = 32,9 dB).* (Echelle : 512*512 pixels)

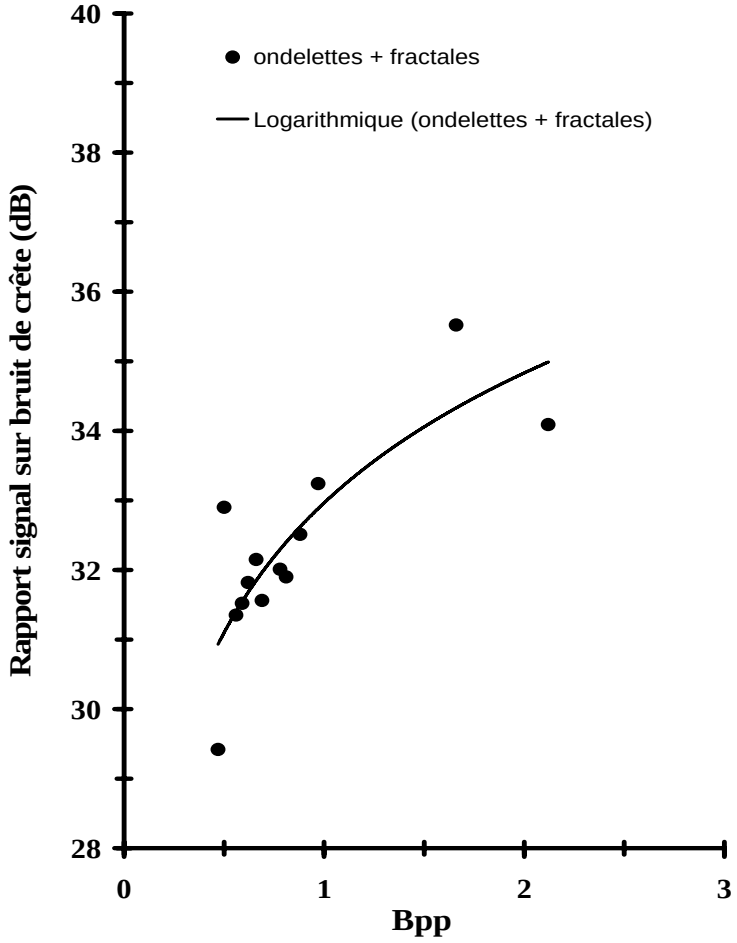


Figure 8 : Rapport signal sur bruit (PSNR) en fonction du taux de compression (bpp) pour l'image de Lena. *Signal to noise ratio (PSNR) against compression ratio (bpp) for the image of Lena.*

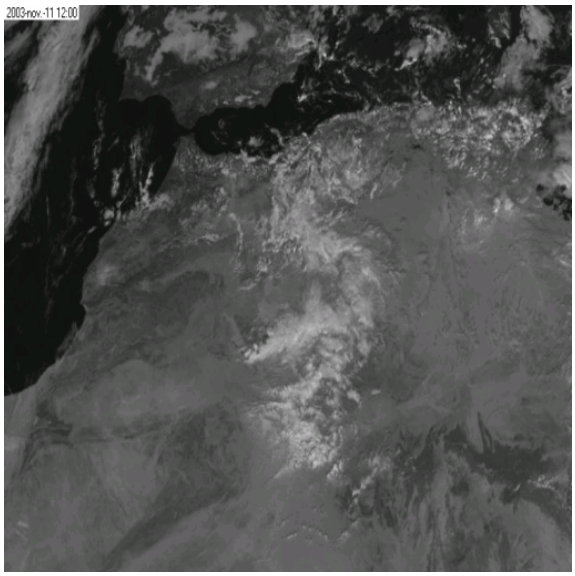


Figure 9 : Reconstitution de l'image HRV de SPOT (bpp = 0,5 bits/pixel et PSNR = 31,05 dB). *Reconstitution of the SPOT HRV image (bpp = 0,5 bits/pixel and PSNR = 31,05 dB).* (Echelle : 512*512 pixels)

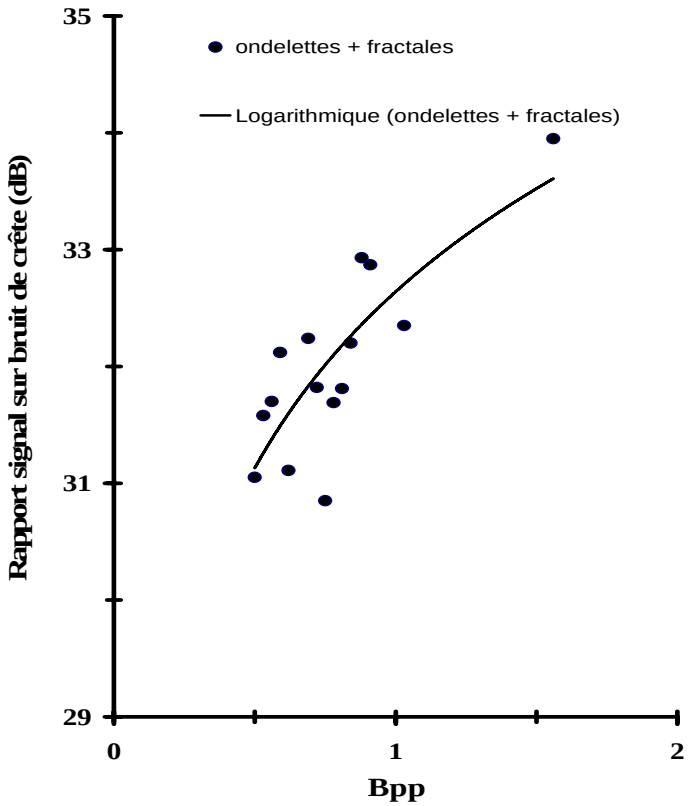


Figure 10 : Rapport signal sur bruit (PSNR) en fonction du taux de compression (bpp) pour l'image HRV de SPOT. *Signal to noise ratio (PSNR) against compression ratio (bpp) for the SPOT HRV image.*

Tableau 1 : Opérations d'isométrie. *Operations of isometry.*

Opérations d'isométrie	**Application à un bloc D_p**
Identité	$I_1\{D_p(i, j)\}=D_p(i, j)$
Réflexion par rapport à l'axe vertical	$I_2\{D_p(i, j)\}=D_p(i,16 – 1 - j)$
Réflexion par rapport à l'axe horizontal	$I_3\{D_p(i, j)\}=D_p(16 – 1 – i , j)$
Réflexion par rapport à la première diagonale	$I_4\{D_p(i, j)\}=D_p(j, i)$
Réflexion par rapport à la seconde diagonale	$I_5\{D_p(i, j)\}=D_p(16 – 1 - j,16 – 1 - i)$
Rotation de + 90°	$I_6\{D_p(i, j)\}=D_p(j , 16 – 1 - i)$
Rotation de + 180°	$I_7\{D_p(i, j)\}=D_p(16 – 1 - i,16 – 1 - j)$
Rotation de – 90°	$I_8\{D_p(i, j)\}=D_p(16 – 1 – j , i)$

Tableau 2 : Bandes spectrales des douze canaux du capteur de METEOSAT. *Spectral bands of the twelve channels of the METEOSAT sensor.*

Canal	**Bande spectrale (µm)**
Visible : HRV	0,56 à 0,71
canal 0.6	0,56 à 0,71
canal 0.8	0,74 à 0,88
Proche infrarouge : canal 1.6	1,5 à 1,78
Infrarouge moyen : canal 3.9	3,48 à 4,36
Vapeur d'eau : canal 6.2	5,35 à 7,15
canal 7.3	6,85 à 7,85
Infrarouge : canal 8.7	8,3 à 9,1
canal 9.7	9,38 à 9,94
canal 10.8	9,8 à 11,8
canal 12.0	11 à 13
canal 13.4	12,4 à 14,4

Tableau 3 : Comparaison des résultats obtenus sur l'image de Lena avec ceux publiés dans la littérature. *Comparison of the results obtained from the image of Lena with those published in the literature.*

Types d'algorithme	**PSNR (dB)**	**Taux de compression C (%)**	**Facteur de performance Q**
Ondelettes + fractales sans itération	32,90	93,75	1827,97931
Quantification vectorielle en sous bandes (Ameur *et al.,* 2002)	32,20	93,75	1555,86273
LIFS d'après Jacquin (1993)	31,40	92,50	1276,85544
LIFS d'après (Lepsoy *et al.,* 1993)	32	93,37	1479,81477
	31,50	94,12	1329,48034
LIFS d'après Vines (1993)	26,56	97,56	441,846879
LIFS d'après Thomas and Deravi (1995)	28,89	93,75	726,057935
JPEG (Charier *et al*., 1999)	32,85	94,91	1829,4139
FVQ (Chung-Su, 1998)	32,22	93,75	1563,04426
JPEG2000 (Charier *et al*., 1999)			

Tableau 4 : Comparaison des temps de compression et de décompression évalués pour différentes approches. *Comparison of the compression and decompression times evaluated for different approaches.*

Types d'algorithme	Temps de compression (s)	Temps de décompression (s)
Ondelettes + fractales sans itération	1	1
DCT (JPEG)	5	1
Ondelettes	3	1
Fractales (LIFS)	180	1

5. CONCLUSION

L'algorithme de compression et de décompression des images que nous avons développé est capable de compresser rapidement toute image avec de forts taux de compression et de la restituer avec une qualité appréciable. Ainsi, les tests effectués sur l'image de Lena prise pour référence et sur des images dans le visible acquises par le capteur du satellite METEOSAT ont montré la supériorité de cet algorithme vis-à-vis d'autres méthodes de compression, telles que celles basées sur la DCT, les ondelettes ou les fractales. En effet, celui-ci réalise un très bon compromis entre l'accroissement de la compression d'une image, la limitation du nombre d'altérations et la réduction du temps de calcul. De telles performances ont pu être atteintes grâce à la combinaison d'ondelettes biorthogonales et de fractales que nous avons adoptée pour réaliser le processus de compression et de décompression des images. Ainsi, le processus de quantification vectorielle, suivi du codage par des fractales, a permis d'augmenter le taux de compression de façon significative, surtout lors de la formation du domaine de contraction. Les artefacts, qui résultent de l'accroissement du taux de compression, sont habituellement des effets de blocs, des effets de bords ou des objets flous. Les filtres biorthogonaux que nous avons employés ont permis de réduire sensiblement l'importance de ces artefacts, car ce sont des filtres symétriques à phase linéaire. Le codage fractal étant sans itération, le temps de calcul est fortement réduit. De plus, le procédé de décimation retenu pour construire les domaines de contraction et de collage contribue aussi à diminuer ce temps.

Notre algorithme, appliqué à l'ensemble des images du capteur de METEOSAT enregistrées durant l'année 2003, a permis de réduire de près de 90 % le volume des données satellitaires à sauvegarder, tout en gardant leurs propriétés initiales. Il représente donc un outil puissant, qui rend possible l'archivage d'images météorologiques sur de longues périodes et, par conséquent, l'étude des changements climatiques à long terme.

Les tests effectués sur deux images de télédétection à haute résolution, de HRV de SPOT et d'IKONOS, ont confirmé l'efficacité de notre algorithme et laissent entrevoir la possibilité d'étendre ce dernier à la compression et à la décompression d'autres types d'images satellitaires. Cependant, d'après ces tests, le traitement et l'archivage d'images de taille beaucoup plus grande ne peuvent guère se réaliser à l'aide d'un microordinateur Pentium 4, à cause de l'insuffisance de sa capacité mémoire. Pour de telles tailles, il serait donc intéressant de rechercher un compromis entre la méthodologie utilisée et la puissance des moyens de calcul requis.

Remerciements

Nous tenons à remercier vivement l'Office National de la Météorologie (Dar El Beida) pour avoir mis à notre disposition les images du capteur de METEOSAT ayant servi à notre étude.

Références

Ameur, S., Adane, A. et Lahdir, M. (2002) Compression d'images Météosat en sous bandes par transformation discrète en cosinus et quantification vectorielle. Télédétection, vol. 2, nº 4, p. 255-266.

Antonini, M., Barlaud, M., Mathieu, P. and Daubechies, I. (1992) Image coding using wavelet transform. IEEE Transactions on Image Processing, vol. 1, nº 2, p. 205-220.

Averbuch, A., Lazar, D. and Israeli, M. (1996) Image compression using wavelet transform and multiresolution decomposition. IEEE Transactions on Image Processing, vol. 5, nº 1, p. 4-15.

Bedford, T., Dekking, F.M. and Keane, M.S. (1992) Fractal image coding techniques and contraction operators. Nieuw Archief Voor Wiskunde, vol. 4, nº 10, p. 185-218.

Belloulata, K. (2005) Fast fractal coding of subbands using a non iterative block clustering. Journal of Visual Communication and Image Representation, vol. 16, p. 55-67.

Borko, F. (1995) A survey of multimedia compression techniques and standards. Part I : JPEG standard. Real Time Imaging, vol. 1, p. 49-67.

Charrier, M., Santa Cruz, D. and Larsson, M.C. (1999) JPEG2000, the next millennium compression standard for still images. Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS'99), Florence, June 7-11, 1999, vol. 1, p. 131-132.

Chung-Su, K., Rin-Chun, K. and Sang-Uk, L. (1998) A fractal vector quantizer for image coding. IEEE Transactions on Image Processing, vol. 7, nº 11, p. 1598-1602.

Cosman, P.C., Gray, R.M. and Vetterli, M. (1996) Vector quantization of image Subbands : A survey. IEEE Transactions on Image Processing, vol. 5, nº 2, p. 202-225.

Daubechies, I., Cohen, A. and Feauveau, J. (1992) Biorthogonal bases of compactly supported wavelets. Communications on Pure and Applied Mathematics, vol. 14, p. 485-560.

Davis, G.M. (1998) A wavelet based analysis of fractal image compression. IEEE Transactions on Image Processing, vol. 7, p. 141-154.

Davoine, F. (1995) Compression d'images fixes par fractales basée sur la triangulation de Delaunay. Thèse de doctorat, Institut national polytechnique de Grenoble, Grenoble, 193 p.

Fisher, Y. (1994) Fractal image compression: Theory and Application. Springer-Verlag, New York, 234 p.

Ghosh, S.K, Mukherjee, J., Das, P.P. (2004) Fractal image compression: a randomized approach. Pattern Recognition Letters, vol. 25, p. 1013-1024.

Hamzaoui, R. and Saupe, D. (2000) Combining fractal image compression and vector quantization. IEEE Transactions on Image Processing, vol. 9, nº 2, p. 197-208.

Held, G. and Marshall, R.D. (1996) Data and image compression : tools and techniques. John Wiley and Sons, Chichester, 432 p.

Jacquin, A.E. (1992) Image coding based on a fractal theory of iterated contractive image transformations. IEEE Transactions on Image Processing, vol. 1, p. 18-30.

Jacquin, A.E. (1993) Fractal image coding : A review. Proceedings of IEEE, vol. 81, p. 1451-1465.

Jacobs, E.W., Fisher, Y. and Boss, R.D. (1992) Image compression: A study of the iterated transform method. Signal Processing, vol. 29, p. 251-263.

Lepsy, S., Oien, G.E. and Ramstad, T.A. (1993) Attractor image compression with a fast non iterative algorithm. IEEE International Conference on Acoustic Speech and Signal Processing, 27-30 avril 1993, Minneapolis, vol. 5, p. 337-340.

Lepsy, S., Carlini, P. and Oien, G.E. (1998) On fractal image compression and vector quantization. *In* Y. Fisher (dir.) NATO ASI Series Book : Fractal Image Encoding Analysis, Springer-Verlag, New York, 368 p.

Mallat, S.G. (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, nº 7, p. 674-693.

Marsault, X. (1995) Compression et cryptage des données multimédias. Hermès, Paris, 239 p.

Moureaux, J.M. (1994) Quantification vectorielle algébrique pour la compression d'images. Application à l'imagerie radar à synthèse d'ouverture (SAR). Thèse de doctorat, Université de Nice-Sofia Antipolis, Nice, 130 p.

Thomas, L. and Deravi, F. (1995) Region-based fractal image compression using heuristic search. IEEE Transactions on Image Processing, vol. 4, p. 832-838.

Vines, G. (1993) Signal modeling with iterated function systems. Thèse de doctorat, Georgia Institute of Technology, Atlanta (GA), 120 p.

Wohlberg, B. and De Jager, G. (1999) A review of the fractal image coding literature. IEEE Transactions on Image Processing, vol. 8, nº 12, p. 1716-1728.

Zhao, Y. and Yuan, B. (1994) A hybrid image compression scheme combining block-based fractal coding and DCT. Signal Processing : Image Communication, vol. 8, p. 73-78.