



Compression fractale par partitionnement de Delaunay

J.-M. Chassery, F. Davoine et E. Bertin

Laboratoire TIMC-IMAG, URA CNRS D 1618, GDR TDSI CNRS
CERMO, Université Joseph Fourier, BP 53X, 38041 GRENOBLE Cedex, FRANCE.

RÉSUMÉ

La méthode de compression par fractals présentée dans ce papier repose sur la théorie fractale des L-IFS (Local Iterated Function System). Un partitionnement de l'image en triangles de Delaunay diminue le nombre de transformations affines utiles pour coder l'image. L'aspect adaptatif de la méthode améliore la restitution des contours dans l'image reconstruite.

ABSTRACT

The principle of fractal image coding presented in this paper is based on the theory of L-IFS (Local Iterated Function System). A Delaunay partition of the image support decrease the number of affine transforms used to code the image. The adaptive aspect of the method improves the boundaries restitution in the reconstructed image.

1 Introduction

La compression d'images par fractals est une technique relativement récente puisqu'elle a été proposée pour la première fois par M. Barnsley, en 1988.

Il s'agissait au départ de coder des formes fractales à l'aide d'un nombre très réduit de coefficients. La théorie des IFS (Iterated Function Systems) permet en effet de générer des images fractales simplement, décrites par un ensemble de transformations affines contractantes. Un théorème a ensuite été démontré par M. Barnsley, permettant de résoudre le problème inverse [1].

Le codage des images réelles nécessite une adaptation de la théorie des IFS. Une possibilité est d'exploiter les similarités locales présentes dans l'image. A. Jacquin fut le premier en 1989 à présenter un algorithme automatique de compression d'images réelles par fractals [7]. Celui-ci consiste à partitionner l'image, et à mettre chacun des blocs carrés obtenus en correspondance avec un autre bloc, de taille supérieure, recherché n'importe où dans l'image. On parle dans ce cas d'IFS local, désigné sous le terme L-IFS. La méthode permet une compression de l'ordre de 8 à 15, pour des images $256 * 256$.

Plusieurs chercheurs ont ensuite utilisé les L-IFS pour la compression d'images [8, 2, 6]. Un algorithme un peu différent [9] découpe l'image en blocs carrés, et recherche l'IFS codant chacun des blocs. Il réalise en fait une approximation fractale des blocs de l'image à coder.

Nous présentons dans ce papier des résultats de compression d'images réelles par fractals (L-IFS), en utilisant un partitionnement en triangles de Delaunay, adapté à l'image. Ce partitionnement augmente la qualité de l'image reconstruite, le taux de compression, et diminue les temps de calcul pour le codage et le décodage.

Le chapitre 2 présente la théorie des IFS, L-IFS. L'algorithme de codage-décodage est détaillé dans le chapitre 3. Le chapitre 4 explique comment calculer le taux de compression. Des résultats sur des images $256 * 256$ sont présentés dans le chapitre 5. La dernière partie propose des améliorations possibles de la méthode, et conclue.

2 Bases théoriques

Considérons un espace métrique (X, d) dans lequel une image digitale est un point. La métrique d est une distance que l'on décrira par la suite. Une transformation contractante W dans cet espace est définie de la manière suivante :

$$W : X \rightarrow X$$

$$\exists s < 1 \text{ tel que } \forall p, q \in X, d(W(p), W(q)) \leq s \cdot d(p, q)$$

Le réel s est dans ce cas appelé facteur de contraction de l'opérateur W .

Un tel opérateur contractant possède un unique point fixe p_f dans X , tel que $p_f = W(p_f)$. L'itération de l'opérateur W , partant de n'importe quel point de X , converge vers le point p_f . Plus précisément on a : $\lim_{n \rightarrow \infty} W^n(q) = p_f, \forall q \in X$.

Le problème inverse de la théorie des IFS consiste à retrouver l'opérateur W codant une image. Un théorème essentiel (appelé "théorème du collage") a été pour cela démontré par M. Barnsley. La solution est de voir l'image A à coder comme une approximation du point fixe de la transformation contractante W . Dans ce cas, $A \simeq W(A)$. L'opérateur retourne un résultat proche de l'image originale A , partant d'une image quelconque, comme expliqué précédemment. Le théorème indique l'inégalité suivante :

$$d(A, A_t) \leq \frac{1}{1-s} d(A, W(A)), \text{ où } A_t \text{ est le point fixe de } W, \text{ appelé aussi l'attracteur de } W.$$

Si on minimise la distance entre l'image A à coder et l'image



transformée par l'opérateur W , alors la distance entre A et l'image attracteur de l'opérateur W est petite, en supposant que le facteur de contraction s de W ne soit pas trop proche de 1.

Un L-IFS (Local Iterated Function System) est composé d'une suite de transformations affines contractantes et locales, définies sur l'espace (X, d) : $\{\omega_i : D_i \rightarrow X : i = 1, \dots, N\}$. $\omega_i(A \cap D_i)$ agit dans notre cas sur une partie D_i de l'image A . $A \cap D_i$ signifie la restriction de l'image A à la partie D_i . Le facteur de contraction du L-IFS est défini comme : $s = \text{Max}\{s_i : i = 1, \dots, N\}$, s_i étant le facteur de contraction de la transformation ω_i .

Le L-IFS permet la construction d'un opérateur contractant W défini comme :

$W(A) = \bigcup_{i=1}^N \omega_i(A \cap D_i)$. Ainsi l'opérateur retourne une image formée de l'union de parties transformées de l'image A .

3 Principe du codage - décodage

Le codage d'une image réelle A en utilisant la théorie des L-IFS consiste à construire l'opérateur W tel que $\bigcup_{i=1}^N \omega_i(A \cap D_i)$ soit une très bonne approximation de A .

Il faut pour cela construire une partition sans recouvrement de l'image en N blocs R_i . L'opérateur W réalise les "collages optimaux" de parties D_i de l'image A sur les blocs R_i . Cela signifie que pour chacun des blocs R_i , on recherche dans l'image une autre zone lui étant similaire. En général, l'erreur quadratique (Mean Square Error) est utilisée pour la mesure de cette ressemblance (figure 1).

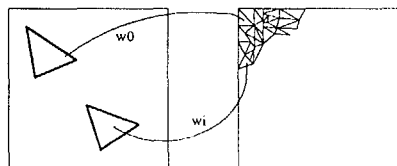


Figure 1: Principe du codage d'une image : Les blocs R_i sont sur l'image de droite. L'image transformée $W(A)$ (droite) doit être très proche de l'image originale A (gauche).

3.1 Partitionnement

Différents types de partitionnements sont possibles [6]. Nous proposons ici un partitionnement de l'image en triangles de Delaunay. Celui-ci a l'avantage de ne pas être rigide. De plus, le diagramme est construit dans un environnement de type division et fusion [4, 3, 5], ce qui lui permet d'être adapté à l'image.

La triangulation de Delaunay d'un ensemble fini de points $T = \{t_n\}$ peut être définie comme étant l'unique triangulation à "cercles vides". Chaque cercle circonscrit à un triangle de Delaunay (t_i, t_j, t_k) ne contient aucun autre point t_l de D . Cette triangulation présente de très bonnes propriétés de régularité.

Le dual géométrique est appelé le diagramme de Voronoi, composé de polygones (figure 2).

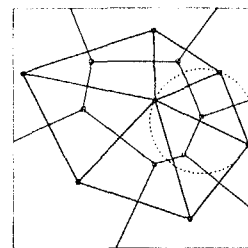


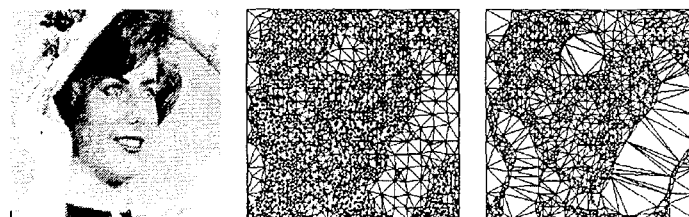
Figure 2: Triangles de Delaunay, et Polygones de Voronoi. Les sommets du polygone constituent les centres des cercles circonscrits aux triangles. Les sommets des triangles constituent l'ensemble T .

3.2 Codage d'une image

3.2.1 Construction des partitions

L'algorithme nécessite une partition R de l'image en triangles de Delaunay. C'est sur cette structure que se font les collages. La recherche des blocs pendant le codage se fait dans une autre partition D , présentant des blocs de surface supérieure, en moyenne. L'utilisation de deux partitions D permet d'avoir des triangles avec recouvrement, donc plus de choix pour réaliser de meilleurs collages.

L'objectif est d'avoir un maximum de ressemblance entre les différents blocs des partitions R et D .



(a) image "femme"

(b) Division : 3476 triangles.

(c) Division et Fusion : 2157 triangles.

Figure 3: Triangulation de Delaunay : partition R .

Pour cela, la triangulation de Delaunay est construite dans un environnement de Division et Fusion. Le diagramme évolue de manière dynamique par rapport à l'information image. La procédure est initialisée avec une distribution régulière de points sur le support de l'image, donnant un nombre limité de triangles. La division consiste à rajouter des points sur le barycentre des triangles dont le niveau de gris n'est pas homogène (critère de variance).

La phase de division est stoppée une première fois pour générer les blocs de grande taille (partition D). Une image $256 * 256$ est ainsi en moyenne recouverte par 500 triangles. La procédure de division est ensuite relancée de manière à obtenir des triangles de plus petite surface (fig. 3b). Elle se poursuit jusqu'à convergence, ou s'arrête si la surface des triangles est inférieure à une valeur limite. Le critère d'arrêt peut aussi être fondé sur le module du vecteur gradient, donnant une information sur la présence de frontières dans les blocs.

La phase de fusion supprime les triangles inutiles, pour les-



quels les voisins ont une valeur moyenne de niveaux de gris très proche. Elle permet d'obtenir des plus grands triangles dans les zones de l'image où le niveau de gris est constant (fig. 3c). Cette étape diminue sensiblement le nombre de triangles dans la partition R. Ceci aura pour effet de diminuer le temps de calcul pour coder l'image ainsi que d'augmenter le taux de compression.

3.2.2 Construction du L-IFS

Une transformation affine ω_i composant le L-IFS est définie comme :

$$\omega_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ o_i \end{pmatrix}.$$

ω_i regroupe deux transformations et peut donc s'écrire aussi $\gamma_i(\varphi_i(A \cap D_i))$ où φ_i est définie par les coefficients $a_i, b_i, c_i, d_i, e_i, f_i$ et γ_i par les coefficients s_i, o_i . φ_i est une transformation affine spatiale. γ_i transforme le niveau de gris des pixels. s_i et o_i contrôlent respectivement la variance et la moyenne des niveaux de gris des pixels.

La transformation spatiale agit linéairement sur un bloc de l'image (rotations, contractions, translations).

Chaque transformation ω_i code la manière de passer d'un bloc D_i de l'image à un autre bloc R_i de taille plus petite, lui ressemblant. Cette opération est désignée sous le terme de collage. La transformation spatiale φ_i est contractante si le module du déterminant $\begin{vmatrix} a_i & b_i \\ c_i & d_i \end{vmatrix}$ est inférieur à 1. Cette condition a pour effet de diminuer la surface de la base du bloc transformé.

Le collage est optimal si la distance (erreur quadratique) entre les blocs est minimale. Celle-ci s'écrit :

$d(A \cap R_i, \omega_i(A \cap D_i)) = \sum_{n=1}^{n_0} (s_i \cdot d_n + o_i - r_n)^2$ où d_n et r_n sont respectivement les intensités des pixel des blocs D_i et R_i , et n_0 le nombre de pixels inclus dans $A \cap R_i$. s_i, o_i sont les coefficients de ω_i .

Dans ce cas, le facteur de contraction de la transformation ω_i est égal à s_i^2 , et ne doit pas être supérieur à 1 pour garantir la convergence de l'opérateur W vers son point fixe, au cours du décodage de l'image.

3.3 Décodage d'une image

Le décodage d'une image se fait en itérant l'opérateur W, à partir d'une image quelconque B (figure 4). L'image reconstruite est théoriquement donnée par : $A \simeq \lim_{n \rightarrow \infty} W^{o_n}(B)$. En pratique le nombre n d'itérations est inférieur à 12. Il dépend du facteur de contraction de l'opérateur W, et de la taille des blocs.

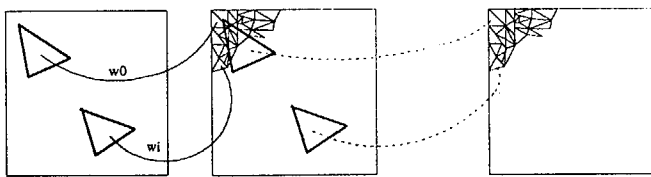


Figure 4: Principe du décodage d'une image : première et dernière itération, partant d'une image quelconque.

Une itération consiste à parcourir les blocs de la partition R (dans le même ordre que celui issu du codage). Les collages des triangles D_i sur les triangles R_i sont réalisés en utilisant les transformations affines contractantes ω_i . Les détails à l'intérieur des blocs R_i vont devenir de plus en plus petits au cours des itérations. La taille de ceux-ci sera limitée par la résolution de l'image.

4 Taux de compression

Si la partition R d'une image possède N blocs, l'image est codée par les N transformations ω_i du L-IFS. Une transformation ω_i est codée par :

- le numéro du triangle D_i dans la structure de données décrivant la partition D,
- la manière de coller le triangle D_i sur le triangle R_i (6 possibilités),
- le coefficient o_i ,
- le coefficient s_i .

Il a été vérifié [6] qu'il suffit de 7 bits pour coder le paramètre o_i et 5 bits pour le paramètre s_i . L'orientation du collage est codée par 3 bits.

La partition D étant construite dans une structure de données de type graphe de Delaunay, il n'est pas utile de coder les positions des germes de chacun des triangles. Un triangle est mémorisé par son numéro dans le graphe. Celui-ci dépend du nombre M de triangles dans la partition D. Il est égal à l'entier supérieur de $\log_2(M)$. En général, le nombre de blocs D_i ne dépasse pas 1024, et donc le codage d'une transformation ω_i nécessite 25 bits.

Les partitions adaptées à l'image doivent de même être mémorisées. Il suffit pour cela de coder les phases de division et fusion, partant d'une distribution régulière de points, donc de triangles de formes homogènes. La division d'un triangle lors de la construction du partitionnement est codée par un 1, la non-division par un 0. On récupère à la fin une suite de 0 et de 1 qui est encore facilement compressible par des techniques classiques de compression. La mémorisation des partitions ne représente donc pas une grande quantité d'information.

La formule donnant le taux de compression d'une image de taille $T * T$ numérisée avec 8 bits/pixel est donc :

$T_c = \frac{T * T * 8}{25 * N + X}$ où N est le nombre de triangles dans la partition R de l'image et X le nombre de bits pour coder les partitionnements de l'image.

5 Résultats

Nous présentons dans ce chapitre des résultats sur des images de taille $256 * 256$, en niveaux de gris, 8 bits par pixel.

Le rapport signal à bruit est généralement utilisé pour mesurer la qualité d'une image décodée.

Celui-ci est donné par la formule :

$RSB = 10 \log_{10} \left(\frac{(2^n - 1)^2}{\frac{1}{n_0} \sum_{i=1}^{n_0} (\mu_i - \nu_i)^2} \right)$ avec n le nombre de bits par pixel de l'image originale, n_0 le nombre de pixels dans l'image, μ_i l'intensité des pixels de l'image originale et ν_i l'intensité des pixels de l'image reconstruite.

L'image "femme" est compressée en utilisant les deux partitionnements présentés en figure 3, (fig.5). Dans le deuxième



cas (fig.5b), le taux de compression est supérieur, alors que la qualité visuelle de l'image reconstruite est restée sensiblement la même. Les erreurs sont visibles dans les zones où les frontières sont très marquées (importantes discontinuités dans les niveaux de gris) car les collages sur ces parties de l'image ne sont pas optimaux. Pour chacun des triangles R_i , le choix de son correspondant est limité. L'image d'erreur montre que les zones à niveau de gris homogène ainsi que les textures sont bien reconstituées.

Une image sera correctement codée par les techniques fractales si elle contient suffisamment de redondance. Des formes, des textures similaires doivent se retrouver à plusieurs endroits différents.

Le temps de compression d'une l'image est compris entre 10 et 25 minutes et la décompression nécessite 10 secondes, sur une Silicon Graphics IndigoTM R4000. Ces temps sont directement liés au nombre de blocs dans la partition R.

6 Discussions et conclusion

Des améliorations peuvent être apportées à la méthode. Une détection de contours permettrait de diminuer les temps de calculs tout en améliorant la qualité de l'image reconstruite. Le choix des correspondants des triangles de la partition R se fait dans ce cas parmi un nombre restreint de triangles. Ceux-ci appartiennent à la même classe, et donc les collages sont plus fiables. En effet, l'erreur quadratique ne permet pas forcément de mettre en correspondance deux blocs avec frontière. Une classification des blocs empêcherait ce type d'erreurs.

La méthode de compression par fractals présentée dans ce papier est comparable à la méthode dite de quantification vectorielle. La principale différence est que la compression par fractals n'a pas besoin d'un code-book universel. Les transformations affines composant le L-IFS sont calculées directement sur l'image, et codent l'image. La phase de reconstruction ne demande rien d'autre que le L-IFS.

Références

- [1] Barnsley M.F., Hurd L.P., *Fractal Image Compression*, AK Peters, Ltd., Wellesley, 1993.
- [2] Beaumont J.M., "Image data compression using fractal techniques", *BT TechNol. J.*, Vol. 9, No. 4, pp 93-109, 1991
- [3] Bertin E., Parazza F., J-M. Chassery, "Segmentation and Measurement based on 3D Voronoi Diagram: Application to Confocal Microscopy", *Computerized Medical Imaging and Graphics*, Vol. 17, pp. 000-008, 1993
- [4] Chassery J.-M., Melkemi M., "Diagramme de Voronoi appliqué à la segmentation d'images et à la détection d'événements en imagerie multi-sources", *Traitement du Signal*, Vol. 8, No. 3, pp. 155-164, 1991
- [5] Chen X., Schmitt F., "Split-and-merge image segmentation based on Delaunay triangulation", *Proc. of The 7th Scandinavian Conf. on Image Analysis, Aalborg, Denmark*, Vol. 2, pp. 910-917, aug.13-16 1991
- [6] Fisher Y., "A Discussion of Fractal Image Compression", in *Chaos and Fractals, New Frontiers of Science*, Peitgen H.O., Jurgens H., Saupe D., Eds. Springer-Verlag, New York, pp. 903-919, 1992
- [7] Jacquin A.E., "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations", *IEEE transactions on image processing*, Vol. 1, No. 1, pp. 18-30, 1992.
- [8] Lepsoy S., Oien G.E., Ramstad T.A., "An Inner Product Space Approach to Image Coding by Contractive Transformations", *ICASSP*, Vol. 3, pp. 2773-2776, 1991
- [9] Monro D.M., Dudbridge F., "Fractal Approximation of Image Blocks", *ICASSP*, Vol. 3, pp. 485-488, 1992



(a) Partitionnement R de la figure 3b. : taux de compression = 6:1 , RSB = 30,4 dB



(b) Partitionnement R de la figure 3c. : taux de compression = 9,5:1 , RSB = 30 dB

Figure 5: Décompressions de l'image "femme". Figures a. et b., de gauche à droite, de haut en bas: image originale, 1^{ère}, 2^{ème}, 3^{ème}, et 10^{ème} itération (image décodée), image d'erreur réhaussée d'un facteur 4.