



# Un partitionnement adaptatif en triangles et quadrilatères pour la compression des images par fractales

Franck Davoine et Jean-Marc Chassery

Laboratoire TIMC-IMAG, Institut Albert Bonniot,  
Domaine de la Merci, 38706 LA TRONCHE Cedex France,  
GDR 134 TDSI CNRS

## RÉSUMÉ

La méthode de compression par fractales présentée dans cet article est basée sur la triangulation de Delaunay. Nous montrons comment, à partir d'une triangulation, diminuer le nombre de blocs en introduisant des quadrilatères. La partition reste adaptée à l'image puisqu'elle présente des triangles de faible taille sur les contours et des quadrilatères de plus grande taille sur les zones homogènes, de moindre importance visuelle. Nous montrons qu'une telle partition permet de conserver une bonne qualité d'image reconstruite à taux de compressions élevés, par rapport aux résultats obtenus à partir d'une triangulation seule.

## ABSTRACT

The fractal image compression algorithm we propose in this article is based on Delaunay triangulation. We show how it is possible to reduce the number of blocks by incorporating quadrilaterals in the partition. The partition remains adapted. It contains small triangles on the contours, and larger quadrilaterals on the homogeneous regions. We show that such a partition preserves a good decoded image quality at high compression ratios, in comparison with a triangulation.

## 1 Introduction

La compression des images numériques, basée sur la géométrie fractale a été proposée pour la première fois par M. Barnsley et A. Sloan en 1988. L'idée est née du constat qu'un *système de transformations itérées* (IFS = Iterated Function System) permet la génération d'objets complexes à l'aide d'un nombre très réduit de coefficients. Les objets sont auto-similaires dans le sens où ils sont composés de l'union de transformations contractantes d'eux-mêmes. Des images naturelles peuvent être approximées en les considérant comme formées de l'union d'objets auto-similaires, chacun codé à l'aide d'un IFS. Les images initialement présentées par M. Barnsley étaient obtenues de cette manière, mais constituaient des approximations très grossières d'images réelles, codées sur un très faible nombre de bits. Le problème inverse revient à déterminer l'IFS codant un objet donné, de manière automatique (sans intervention humaine). Sa résolution est très difficile, mais est indispensable si l'on veut coder une image par un ensemble d'IFS.

A. Jacquin a proposé une méthode automatique permettant le codage d'images naturelles par fractales [5]. Son travail, basé sur l'utilisation d'un système de transformations itérées *locales* constitue le point de départ de nombreuses autres recherches sur la compression des signaux réels 1D, 2D et 3D par fractales [5] [4] [7]. Celles-ci concernent principalement (dans le cas du codage d'images naturelles) :

- la construction d'une partition adaptative pour calculer la transformation fractale
- l'accélération de l'étape de codage.
- la recherche de fonctions élémentaires nouvelles (affines ou autres) pour le codage efficace de la redondance spatiale de l'image.

- l'ajout de vecteurs fixes pour le calcul des transformations itérées.
- l'accélération du décodage : itératif, non-itératif, hiérarchique
- les études théoriques sur le contrôle de la convergence du décodeur
- l'extension de la méthode au codage des images vidéo.

Dans cet article, nous présentons un codeur utilisant deux partitions. Notre but est d'améliorer la qualité visuelle de l'image reconstruite par l'utilisation de partitionnements souples et adaptatifs en triangles de Delaunay. Le taux de compression de la méthode est contrôlé par l'introduction de quadrilatères dans la partition, en fusionnant des triangles voisins.

## 2 Brefs rappels théoriques

### 2.1 Codage par Systèmes de transformations Itérées (IFS)

Nous rappelons brièvement dans cette partie les principaux aspects de la théorie des IFS permettant le codage et la synthèse d'images *binaires* formées d'objets auto-similaires. Considérons l'espace métrique  $(H(\mathbb{R}^2), h(d))$  dont les éléments sont les sous-ensembles compacts non vides de  $\mathbb{R}^2$ . La distance  $h(d)$  est la distance de Hausdorff, fonction de la distance Euclidienne  $d$ .

Soit  $\omega : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  une transformation contractante sur l'espace  $(\mathbb{R}^2, d)$ . Il est montré [1] que dans ce cas, la transformation  $\omega : H(\mathbb{R}^2) \rightarrow H(\mathbb{R}^2)$  définie par  $\omega(B) = \{\omega(x) : x \in B\}$ ,  $\forall B \in H(\mathbb{R}^2)$  est contractante sur  $(H(\mathbb{R}^2), h(d))$ . L'IFS défini par M. Barnsley est composé d'un ensemble de telles transformations contractantes  $\omega_i (i = 1, \dots, N)$  opérant sur des objets inclus dans  $H(\mathbb{R}^2)$ .



Soit un IFS  $\{\mathbb{R}^2; \omega_i, i = 1, \dots, N\}$ . Le nouvel opérateur  $W : H(\mathbb{R}^2) \rightarrow H(\mathbb{R}^2)$  défini par :

$$W(B) = \bigcup_{i=1}^N \omega_i(B), \quad \forall B \in H(\mathbb{R}^2)$$

est contractant et possède un unique *point fixe*  $\mathcal{A}_t = W(\mathcal{A}_t)$  donné par :

$$\mathcal{A}_t = \lim_{n \rightarrow \infty} W^{on}(X), \quad \forall X \in H(\mathbb{R}^2)$$

(la notation  $W^{on}$  signifie que l'opérateur  $W$  est itéré  $n$  fois). L'objet  $\mathcal{A}_t$  est aussi appelé *attracteur* de l'IFS.

Un théorème essentiel (théorème du collage généralisé) [4] fournit une borne supérieure à la distance entre une image originale et l'attracteur d'un opérateur  $W$  :

$$h(A, \mathcal{A}_t) \leq \frac{1}{1-s} \frac{1-\sigma^n}{1-\sigma} h(A, W(A))$$

où  $W$  est un opérateur finalement contractant ( $W$  n'est pas contractant mais l'est à l'itération  $n$ ),  $\sigma$  est le facteur de Lipschitz de  $W$  et  $s$  est le facteur de contraction de  $W^{on}$ . L'opérateur  $W$  peut être finalement contractant si un nombre limité de transformations  $\omega_i$  ne sont pas contractantes.

Le théorème dit que s'il est possible de transformer un objet  $A$  de manière à vérifier  $A \simeq W(A)$  et de s'assurer que  $W$  est finalement contractant, alors le point fixe  $\mathcal{A}_t$  de l'opérateur  $W$  est proche de  $A$ . L'opérateur  $W$  caractérise dans ce cas complètement l'approximation  $\mathcal{A}_t$  de l'objet  $A$ , et code ce dernier de manière exacte si  $A = W(A)$ . Le codage sera d'autant plus efficace que l'objet  $A$  est auto-similaire.

## 2.2 La Transformation Fractale

La transformation fractale est basée sur les notions introduites dans la partie 2.1. L'espace métrique sur lequel elle s'opère est celui des images numériques en niveaux de gris. La distance  $d$  utilisée est, pour des raisons de rapidité de calculs, la distance quadratique entre niveaux de gris.

Le principe de la transformation fractale repose sur le fait qu'une image naturelle  $A$  est "auto-similaire par parties" : des zones de l'image sont mises en correspondance avec d'autres zones de tailles plus petites, leur ressemblant. Le but est de construire un opérateur  $W$  de manière à ce que l'image transformée  $W(A)$  reste quasiment inchangée. L'opérateur  $W$  doit être finalement contractant pour assurer la convergence de ses itérations successives vers un point fixe. Ce point sera proche de l'image originale  $A$  puisque l'on vérifie  $A \simeq W(A)$ .

Une solution proposée par A. Jacquin [5] est de partitionner l'image en  $N$  blocs  $R_i$  :

$$A = \bigcup_{i=1}^N (R_i), \quad R_i \cap R_j = \emptyset \quad \forall i, j$$

et de "coller" sur chacun d'eux une autre partie transformée  $\omega_i(D_i)$  de l'image, de manière à vérifier  $A \simeq W(A)$ . L'image transformée devient :

$$W(A) = \bigcup_{i=1}^N \omega_i(D_i). \quad (1)$$

## 3 Partitionnements de Delaunay

Le principe des algorithmes classiques de compression par fractales est de "capturer" la redondance visuelle *locale* à l'intérieur de l'image à l'aide de transformations affines contractantes. La transformation fractale est directement calculée sur une partition  $R$  dont les propriétés recherchées sont les suivantes :

- grande adaptabilité à l'image (positionnement non rigide des blocs sur le support de l'image)
- localisation facile des blocs dans la partition (relations de voisinage, programmation dans un graphe)
- information minimale nécessaire à son codage

Nous préconisons pour toutes ces raisons la triangulation de Delaunay, calculée sur un ensemble de points bien placés sur le support de l'image, tenant compte du niveau de gris des pixels.

La triangulation de Delaunay d'un ensemble  $S$  de sommets (points)  $p_i$  est définie comme l'unique triangulation à cercles vides, c'est à dire que chacun des cercles circonscrits aux triangles est vide de tout autre sommet. La triangulation est la plus "équilatérale" possible. Différentes approches sont possibles pour calculer la triangulation d'un ensemble de points. Nous utilisons un algorithme incrémental, programmé dans une structure de graphe, consistant à modifier localement la triangulation, après insertion ou suppression d'un nouveau point [2].

La partie 3.1 présente une méthode pour positionner des points sur le support de l'image, de manière à calculer une triangulation adaptée (voir figure 3.2a).

### 3.1 Algorithme de Divisions - Fusion

L'algorithme procède en deux étapes : une ou plusieurs phases de division des triangles, suivies d'une phase de fusion [2]. L'algorithme est initialisé sur une triangulation régulière. Une phase de division consiste à insérer un nouveau point sur le barycentre de chacun des triangles non homogènes (critère sur l'écart type des niveaux de gris des pixels inclus dans les triangles), puis à trianguler le nouvel ensemble de points. Plusieurs phases de division peuvent être nécessaires si un nombre trop important de triangles restent non homogènes. La phase de fusion consiste à supprimer les points inutiles de la triangulation (critère sur la différence de moyenne et de variance des niveaux de gris de chaque triangle ayant ces sommets en commun).

### 3.2 Extraction de quadrilatères

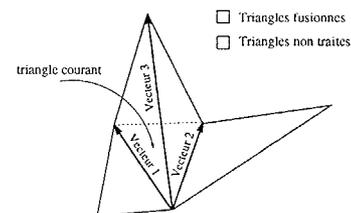
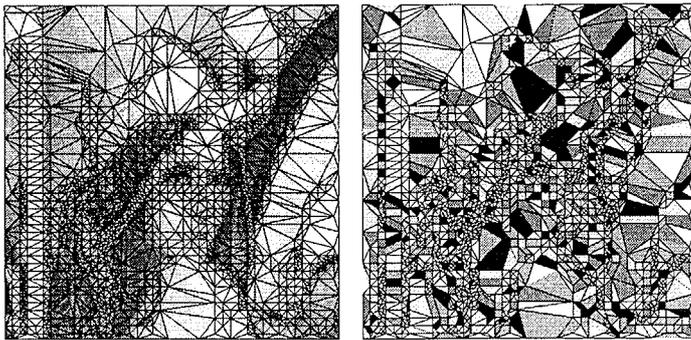


Figure 1 - Regroupement de deux triangles dans un quadrilatère.

Dans le but de réduire le nombre de blocs dans la triangulation de Delaunay, il est possible de regrouper des triangles

voisins, de manière à former des quadrilatères. Un regroupement se fait à condition que les deux triangles candidats ne soient pas sur un contours de l'image, et que leur union génère un bloc convexe.

La détection de la convexité du bloc issu de l'union de deux triangles se fait par l'étude des signes des deux produits vectoriels  $Vecteur1 \wedge Vecteur3$  et  $Vecteur2 \wedge Vecteur3$  (voir figure 1). Les vecteurs  $Vecteur1$  et  $Vecteur2$  sont choisis sur un point d'origine qui appartient au triangle courant et qui n'est pas sur l'arête commune aux deux triangles étudiés. Le vecteur  $Vecteur3$  relie ce même point au sommet opposé, par rapport à l'arête commune (vecteur diagonal). Si les produits vectoriels sont de signes différents, le quadrilatère est convexe. Les deux triangles sont fusionnés dans le quadrilatère. Si les produits vectoriels sont de mêmes signes, le quadrilatère est concave. Dans ce cas, la procédure de détection est relancée sur un autre triangle, voisin du triangle courant. Cet algorithme rapide permet en outre de fusionner les triangles "étirés" créés par l'étape de fusion (§3.1). Il retourne des quadrilatères sur les zones homogènes, ou faiblement texturées, et préserve les triangles sur les contours (voir figure 3.2b).



(a) Divisions - Fusion

(b) Triangles + quadrilatères

Figure 2 - (a): 3136 triangles, (b): partition composée 1706 triangles et de 715 quadrilatères.

## 4 Codage - Décodage

### 4.1 Codage

La phase de codage consiste à mettre en correspondance chacun des  $N$  blocs  $R_i$  d'une partition  $R$  avec une autre partie transformée  $\omega_i(D_i)$  (eq. (1)) de l'image à coder, minimisant la distance quadratique  $d(R_i, \omega_i(D_i))$ . Notre approche est basée sur une triangulation  $R$  adaptée à l'image, fournissant un nombre minimal de blocs. Les blocs  $D_i$  associés à chacun des blocs  $R_i$  devraient être recherchés dans l'image entière. Nous nous contentons, en pratique, de les rechercher dans une deuxième partition  $D$ , plus grossière et non adaptée à l'image. La partition  $D$  fournit un ensemble de blocs de taille, en moyenne, supérieure à celle des blocs  $R_i$ .

La transformation  $\omega_i$  transformant un bloc  $D_i$  en un bloc  $R_i$  s'écrit :

$$\begin{aligned} \omega_i(D_i) &\equiv \omega_i(x_m, y_m, f(x_m, y_m)) \quad \forall (x_m, y_m) \in D_i \\ &= (x'_m, y'_m, f(x'_m, y'_m)) \end{aligned}$$

$$= (v_i(x_m, y_m), s_i f(x_m, y_m) + o_i),$$

où  $f(x_m, y_m)$  est la luminance du pixel de coordonnées  $(x_m, y_m)$  à l'intérieur du bloc  $D_i$ ,  $f(x'_m, y'_m)$  est la luminance du pixel de coordonnées  $(x'_m, y'_m)$  à l'intérieur du bloc  $R_i$ , et les coefficients  $s_i, o_i$  contrôlent respectivement le contraste et la luminosité des niveaux de gris. La transformation spatiale  $v_i$  permet le collage d'un bloc  $D_i$  sur un bloc  $R_i$ .

• Le collage d'un triangle sur un autre se fait à l'aide d'une transformation  $v_i$  affine:

$$v_i(x_m, y_m) = ((a_i x_m + b_i y_m + e_i), (c_i x_m + d_i y_m + f_i))$$

Les coefficients  $a_i \dots f_i$  sont calculés en considérant les trois sommets du triangle de départ et les trois sommets du triangle d'arrivée.

• Le collage d'un quadrilatère sur un autre se fait à l'aide d'une transformation  $v_i$  donnée par :

$$v_i \begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} x'_m \\ y'_m \end{bmatrix} = \begin{bmatrix} a_m \\ b_m \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix}$$

Le calcul des quatres transformations opérant sur les sommets des quadrilatères est simple :

$$\begin{bmatrix} x'_k \\ y'_k \end{bmatrix} = \begin{bmatrix} a_k \\ b_k \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad \forall k = 1 \dots 4$$

Les coefficients de la transformation d'un point  $(x_m, y_m)$  à l'intérieur du bloc  $D_i$  sont ensuite calculés par interpolation bilinéaire :

$$\begin{bmatrix} a_m \\ b_m \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \end{bmatrix} \begin{bmatrix} (1-\alpha)(1-\beta) \\ \alpha(1-\beta) \\ \beta(1-\alpha) \\ \alpha\beta \end{bmatrix}$$

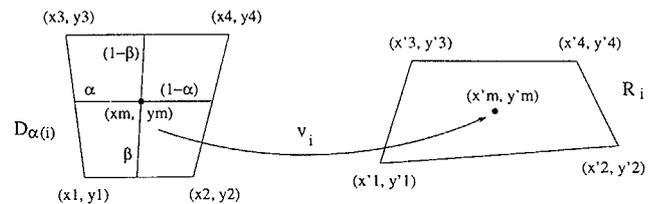


Figure 3 - transformation d'un quadrilatère.

En pratique, le calcul de la distance  $d(R_i, \omega_i(D_i))$  considère tous les pixels inclus dans le bloc  $R_i$ , et les pixels issus du sous-échantillonnage du bloc  $D_i$ . Pour cela, chaque pixel du bloc  $R_i$  est comparé au pixel le plus proche de son antécédant à l'intérieur du bloc  $D_i$  (sous la transformation spatiale  $v_i^{-1}$ ). La distance est donnée par :

$$d(R_i, \omega_i(D_i)) = \sum_{m=1}^{n_r} (f(x'_m, y'_m) - s_i f[v_i^{-1}(x'_m, y'_m)] - o_i)^2$$

pour tout  $(x'_m, y'_m)$  inclus dans  $R_i$ . Le facteur de contraction de l'opérateur  $W$  étant égal à  $s = \sup\{s_i\}$  pour  $i = 1 \dots N$ ,  $W$  est finalement contractant lorsque  $0 < s < 1$ . On montre expérimentalement que  $W$  reste finalement contractant lorsque  $s < 1.5$  [4].

Les temps de calculs sont importants pour le codage d'une image (2 à 10 minutes pour une image  $256 \times 256$ ). Ceux-ci



sont principalement dûs à la manipulation de blocs de formes quelconques, disposés sur une matrice discrète rectangulaire. La phase de codage peut être sensiblement accélérée par une classification des triangles avant codage, dans le but de diminuer le nombre de comparaisons inutiles entre blocs. La classification est simple lorsque la triangulation est contrainte par les contours de l'image. Une autre solution consiste à ne conserver dans la triangulation  $D$  qu'un faible nombre de triangles "visuellement" différents [3]. Cette étape est réalisée par quantification des vecteurs "histogrammes" des valeurs de niveaux de gris de chacun des blocs  $D_i$ . Au cours du codage, chaque triangle  $R_i$  de la triangulation  $R$  n'est comparé qu'au seuls triangles retenus dans la partition  $D$ .

## 4.2 Compression

Le codage de la transformation fractale revient à mémoriser :

- la partition  $R$
- l'information nécessaire au codage des  $N$  transformations  $\omega_i$  composant l'opérateur  $W$ . Chacune d'elle comprend :
  - la position du bloc  $D_i$  à transformer (associé au bloc  $R_i$ )
  - la manière de coller un bloc sur un autre bloc (il existe 6 (resp. 8) solutions différentes pour coller un bloc  $D_i$  sur un bloc  $R_i$ , lorsque ces derniers sont des triangles (resp. quadrilatères))
  - le coefficient  $s_i$  (facteur d'échelle sur les niveaux de gris) et le coefficient  $o_i$  (translation sur les niveaux de gris)

Les coefficients de la transformation spatiale  $v_i$  ne sont pas mémorisés puisqu'ils peuvent être calculés par le décodeur, connaissant la forme des blocs.

En pratique, une transformation  $\omega_i$  est codée sur 25 bits. Un algorithme efficace de codage de la partition  $R$  consiste à ne coder que les décisions de division et de fusion des triangles, en partant d'un lattice régulier et connu. La triangulation de la figure 3.2a est, par exemple, codée sur 5580 bits. La partition régulière  $D$  n'est codée que par le pas du lattice à partir duquel elle est calculée.

## 4.3 Décodage

Le décodage consiste à itérer l'opérateur  $W$ , après reconstruction des partitions  $R$  et  $D$ , en partant d'une image quelconque  $f_0$ . Le niveau de gris  $f_n(x'_m, y'_m)$  du pixel inclus dans le bloc  $R_i$ , à la  $n^{\text{ième}}$  itération de l'opérateur  $W$ , est donnée par :

$$f_n(x'_m, y'_m) = s_i f_{n-1} [v_i^{-1}(x'_m, y'_m)] + o_i \quad \forall (x'_m, y'_m) \in R_i.$$

Le résultat converge en pratique au bout de 5 à 10 itérations vers l'image reconstruite, attracteur de la transformation fractale. Le nombre d'itérations dépend essentiellement du rapport de surface entre les blocs de la partition  $D$  et ceux de la partition  $R$ . Au cours des itérations, le collage d'un grand bloc  $D_i$  (recouvrant plusieurs blocs  $R_k$ ) sur le bloc  $R_i$  a pour effet de contracter les motifs à l'intérieur des blocs de la partition  $R$ . Plus la différence de taille entre les blocs  $D_i$  et les blocs  $R_i$  est importante, plus la convergence est rapide. Un

compromis doit être fait car dans ce cas les blocs sont moins semblables, et le point fixe de l'opérateur  $W$  (image reconstruite) peut s'éloigner de l'image originale. Une étude sur un cas particulier (partitionnement carré) est donnée dans [6].

## 5 Résultats et conclusion

La partition de Delaunay, calculée sur un ensemble de points bien positionnés sur le support de l'image, fournit un nombre restreint de blocs, permettant le décodage d'une image de qualité visuelle acceptable. Par rapport au codage sur une partition triangulaire seule, l'incorporation de quadrilatères permet soit

- d'augmenter le taux de compression de la méthode en fusionnant des triangles dans la triangulation initiale, et en diminuant ainsi le nombre de blocs à traiter, (la qualité visuelle de l'image reconstruite est dans ce cas moins bonne)
- d'améliorer la qualité de l'image reconstruite, à taux de compression constant, en affinant la triangulation initiale (avant extraction des quadrilatères).

La table 1 confirme le dernier point. L'évaluation numérique des résultats est donnée par le rapport signal à bruit de crête (Peak SNR) entre l'image décodée et l'image originale. Nous pouvons remarquer que ce dernier décroît moins vite, à taux de compression élevé, lorsqu'une combinaison triangles + quadrilatères est utilisée. Les faibles différences de qualité visuelle sont principalement perceptibles sur les contours.

	triangles seuls	triangles + quadrilatères
$T_c = 24.7:1$	30.14 dB	29.95 dB
$T_c = 31.6:1$	29.16 dB	29.16 dB
$T_c = 34.9:1$	28.80 dB	29.00 dB

TAB. 1 - Résultats de décodage sur l'image LENA ( $512 \times 512$ ) : rapport signal à bruit de crête  $PSNR$  pour différents taux de compression  $T_c$ . Les partitions donnant les résultats de la deuxième ligne sont celles de la figure 2.

## Références

- [1] M. F. Barnsley and L.P. Hurd. *Fractal Image Compression*. AK Peters Ltd., Wellesley, 1993.
- [2] E. Bertin. Diagrammes de voronoï 2D et 3D : Applications en analyse d'images. *Thèse de l'Université Joseph Fourier*, 1994.
- [3] F. Davoine, M. Antonini, J. M. Chassery, and M. Barlaud. Fractal image compression based on delaunay triangulation and vector quantization. *to appear in the IEEE Transactions on Image Processing - Special Issue on Vector Quantization*, 1995.
- [4] Y. Fisher (Ed.). *Fractal Image Compression: Theory and Application to Digital Images*. Springer Verlag, New York, 1995.
- [5] A. E. Jacquin. Fractal image coding: A review. *Proceedings of the IEEE*, 81(10):1451-1465, October 1993.
- [6] G. E. Øien and S. Lepsøy. Fractal-based image coding with fast decoder convergence. *Signal Processing*, 40:105-117, October 1994.
- [7] D. Saupe and R. Hamzaoui. A review of the fractal image compression literature. *Computer Graphics*, 28(4):268-276, 1994.