

# Compression d'images couleurs par hybridation d'un réseau de neurones et fractale

S. ELHANNACHI<sup>1</sup> N.BENAMRANE<sup>1</sup>

<sup>1</sup> Université des Sciences et de la Technologie d'Oran, Mohamed Boudiaf, Département d'Informatique, BP 1505 EL-Mnaouer, 31000 Oran, ALGERIE

sidahmed\_elhannachi@yahoo.fr nabenamrane@yahoo.com

A.FREVILLE<sup>2</sup>

<sup>2</sup> LAMIH - Université de Valenciennes, CNRS UMR 8530 Le Mont Houy F59313 Valenciennes cedex 9, FRANCE

Arnaud.Freville@univ-valenciennes.fr

**Résumé** – En compression Fractale l'étape de codage nécessite un coût calculatoire important. Dans ce papier, un schéma de codage rapide pour la compression d'image par fractale, est proposé. La technique est basée sur l'hybridation des fractales et la quantification vectorielle par un réseau de Kohonen à plusieurs dictionnaires. Cette technique a été testée sur des images couleurs. Elle a sensiblement diminué le temps de calcul du codeur fractal, tout en maintenant une qualité satisfaisante des images reconstruites à des taux de compression importants.

**Abstract** – In Fractal compression the stage of coding requires a significant computational cost. In this paper, a fast coding schema for the fractal image compression is proposed. The technique is based on the hybridization of the fractals and the Vector Quantization by Kohonen's network with multiple code books. This technique was tested on color images. It appreciably decreased the computing time of the fractal coder, while maintaining a satisfactory quality of the reconstructed images at significant compression ratios.

## 1. Introduction

La compression d'images est devenue une tâche essentielle pour faire face à la quantité croissante d'informations que l'on souhaite transmettre ou stocker.

A l'aide des techniques de compression, le stockage et la transmission des images seront plus efficaces et plus rapides. La grande variété des domaines d'exploitation, chacun ayant ses contraintes spécifiques, conduit aujourd'hui à un très grand nombre de procédés de compression. En effet, les techniques de compression se divisent en deux catégories principales : compression sans perte et compression avec perte. Parmi les techniques sans perte, nous citons : la norme JPEG, la quantification vectorielle, les ondelettes et JPEG 2000[1].

La compression fractale fait partie des méthodes de compression d'images irréversibles (avec perte). Elle est devenue l'un des champs privilégiés d'investigation en compression d'images numériques. Depuis que Jacquin a présenté un schéma pratique de codage fractal d'image par bloc [2] en 1990, cette méthode a fait éveiller beaucoup d'attention [2][3], comme étant une technique prometteuse de compression d'image.

La compression fractale est basée sur les IFS (systèmes de fonctions itérées) qui exploitent l'auto-similarité entre les parties de l'image. En effet l'image est partitionnée en blocs  $R_i$  (Blocs destination). Chacun de ces blocs est ensuite mis en correspondance avec un autre bloc  $D_i$  de l'image (Bloc source), à partir duquel il est possible d'approximer un bloc  $R_i$ , par une transformation élémentaire.

Divers travaux ont montré que la méthode de compression par fractal possède un potentiel lui permettant de figurer parmi les méthodes de compression efficace [3][4]. Cependant la durée de la phase de codage est importante. Cela est essentiellement dû au grand nombre de comparaisons entre les blocs destination de la partition R et les blocs source de la partition D.

La classification des blocs source et destination est l'une des principales approches de réduction du temps de calcul. Dans la méthode proposée par Fisher [3], un bloc donné de l'image, est divisé en quatre sous-blocs. Pour chaque sous-bloc, la moyenne et la variance sont calculées, et selon certaines combinaisons de ces valeurs, 72 classes sont construites. Cette méthode réduit efficacement l'espace de recherche, cependant, elle exige une importante

charge de calculs, l'appariement des 72 classes est assez compliqué.

Dans les travaux de R.Hamzaoui [5], les blocs source sont classifiés en regroupant leurs vecteurs caractéristiques en cellules de Voronoi dont les centres sont issus de l'image de test ou d'un ensemble d'images d'apprentissage.

D'autres méthodes de réduction de complexité, basées sur la réduction du nombre des blocs source de l'ensemble, ont été proposées. D. Monro [11] localise l'ensemble D des blocs source relativement au voisinage du bloc destination courant, fondée sur l'hypothèse que des blocs source qui sont voisins au bloc destination concerné, sont bien admis pour l'assortiment de ce bloc. M.Hassaballah [6] a proposé une nouvelle méthode pour réduire la durée de codage de la compression fractale. Lors du codage fractal, un nombre important de blocs ayant une entropie élevée, ne sont pas utilisés. Ces blocs inutiles sont éliminés de l'ensemble D, réalisant ainsi un ensemble plus productif de blocs source. D'autres approches de réduction du temps de calcul ont été proposées [7][8] et [9].

Dans ce papier, nous proposons une technique destinée à réduire rigoureusement la complexité de temps de l'algorithme.

La section 2, décrit brièvement le codage d'image par fractal. Dans la section 3, l'algorithme de Kohonen et son application à la quantification vectorielle sont présentés. La section 4 présente l'approche proposée, suivie des résultats expérimentaux et de discussions dans la section 5 Une conclusion ainsi que quelques perspectives sont données dans la dernière section.

## 2. Principe du codage fractal

La compression par fractale est fondée sur la recherche d'auto-similarités dans l'image. Elle consiste à approximer un bloc de l'image (appelé bloc destination) par un bloc de taille supérieure (appelé bloc source). En effet l'image est d'abord divisée en blocs non recouvrants  $R_i$  (Blocs destination), d'une taille prédéfinie  $B \times B$ . Puis, un codebook de recherche est créé à partir de l'image prenant tous les blocs carrés  $D_i$  (Blocs source) de la taille double  $2B \times 2B$ .

L'approximation est réalisée selon une transformation affine contractante du bloc source sur un bloc destination. La propriété de contraction se situe aussi bien au niveau spatial (Transformation géométrique) qu'au niveau des intensités (Transformation massique). L'union de telles transformations contractantes, appelée "système de fonctions itérées" (IFS en anglais) permettra d'assurer la convergence d'un processus itératif vers un point fixe, appelé attracteur, proche de l'image à coder [3].

### 2.1 Transformation géométrique

C'est une opération de décimation du bloc à l'aide d'une opération de moyennage. Elle consiste également à appliquer des opérations de rotations (Identité, rotation de  $\pi/2$ ,  $\pi$ ,  $3\pi/2, \dots$ ).

### 2.2 Transformation massique

La fonction de collage massique prend pour expression :

$$\check{R}_i = s \times \check{D}_i + o \quad (1)$$

où  $s \in \mathcal{R}$  et  $o \in N$  sont appelés respectivement facteur d'échelle ou *Scale* et facteur de décalage ou *offset*. Elle contient un ajustement de contraste et un décalage de niveaux de gris afin d'affiner l'approximation du bloc  $R$ .

## 3. Quantification vectorielle par réseau de Kohonen :

Avec les cartes auto-organisatrices de Kohonen, on dote le dictionnaire d'une structure topologique permettant d'améliorer le système.

Le réseau neuronal de Kohonen est connu sous le nom de carte auto-organisatrice (*Self-Organizing Feature Map, SOM*). La séquence d'entraînement contient les blocs des images d'apprentissage, et après l'entraînement du réseau les vecteurs de poids représentent les mots de code finaux.

### 3.1 Carte topologique de Kohonen

La carte topologique de Kohonen est un réseau de neurones à une seule couche où les neurones sont organisés généralement sur une matrice. Chacun de ces neurones étant relié à l'entrée par l'intermédiaire de poids synaptiques variables [10].

L'apprentissage s'effectue itérativement, en présentant chaque vecteur  $x$  de l'ensemble d'entraînement. Les neurones calculent alors la distance (distance euclidienne) entre leur vecteur  $w_i$  et  $x$ , et celui qui est le plus proche est choisi. Les poids du neurone gagnant (désigné par  $s$ ), ainsi que ceux de ses voisins, subissent alors une modification selon l'équation (2) :

$$w_r^{t+1} = w_r^t + \mu h_{rs} \cdot (x - w_r^t) \quad (2)$$

où :

- $\mu$  représente un taux d'apprentissage, généralement fonction du temps :  $\mu = \mu(t)$ ,
- $h_{rs}$  est fonction de la distance entre le neurone  $r$  et le neurone actif  $s$  :  $h_{rs} = h(r, s, t)$ ,
- $t$  est le nombre d'itérations effectuées (temps).

L'équation (2) de mise à jour réalise en fait un déplacement du vecteur de poids vers le point  $x$ . L'importance de ce déplacement est gérée par le facteur  $\mu$  qui évolue dans l'intervalle  $[0,1]$  et décroît en temps. L'étendue du voisinage  $h_{rs}$  est aussi inversement proportionnelle au nombre d'itération.

## 4. Approche proposée

Notre approche est basée sur le schéma de compression fractale développé par Jacquin [1]. L'inconvénient majeur de cette méthode de compression concerne le temps de codage. Cela est essentiellement dû au grand nombre

d'appariements entre les blocs destination de la partition R et les blocs source de la partition D.

Nous proposons une méthode destinée à accélérer la phase de compression fractale en associant le codage par fractales et la quantification vectorielle. On présente deux techniques permettant d'accélérer la phase de codage : la classification des blocs et la quantification des blocs source.

#### 4.1 Classification des blocs

La classification se base sur l'utilisation d'une méthode de détection de contours. Dans notre approche, nous avons utilisé l'algorithme de croissance de régions. Les étapes de cette classification sont :

- Détecter les contours qui existent dans l'image
- Déterminer les blocs incluant des contours.

Les autres blocs (n'incluant pas de contours) seront séparés en deux autres classes (blocs homogènes, blocs texturés) en appliquant un seuil sur la variance du bloc.

Ceci est dans le but de réduire le nombre de comparaisons entre les blocs destination et les blocs source, en se basant sur le fait qu'un bloc destination ne sera comparé qu'avec les blocs source de même classe.

#### 4.2 Quantification des blocs source

Dans cette technique, on propose de quantifier les blocs sources à l'aide d'un quantificateur vectoriel de façon à ne conserver dans la partition D qu'un nombre réduit de blocs représentatifs de l'image. Ceci permet pour chaque bloc destination d'accélérer la recherche du bloc source correspondant et par conséquent de construire la transformation contractante associée.

Dans notre approche, nous effectuons donc une quantification vectorielle à l'aide d'un réseau de Kohonen multi-dictionnaires. Cette méthode s'appuie sur le partitionnement du dictionnaire (la carte topologique du réseau de Kohonen) de taille importante en plusieurs sous-dictionnaires de taille restreinte.

En effet, lors de la quantification (selon un l'algorithme d'apprentissage non supervisé de Kohonen), les blocs de l'image utilisée sont tout d'abord classifiés selon le contenu dans chaque bloc, ainsi, plusieurs dictionnaires seront créés selon le nombre de classes et chaque dictionnaire sera spécialisé dans la quantification d'un type bien défini de blocs. Dans notre cas, le dictionnaire est divisé en deux sous-dictionnaires :

- Un dictionnaire pour les blocs contours,
- Un autre pour les blocs texturés.

Lors du codage fractal, on ne considère qu'un nombre restreint de blocs source au sein de la partition D, en utilisant les blocs quantifiés des deux dictionnaires sachant que pour un bloc destination, la recherche du bloc source associé, n'est effectué que dans les blocs du dictionnaire de même classe.

On peut à présent donner l'algorithme d'un codeur Fractal amélioré par une quantification vectorielle par réseau de Kohonen :

1. Détermination des blocs Destination (Partition R) et des blocs Source (Partition D)
2. Classification des blocs (Blocs : Homogènes, Texturés, Contours).
3. Quantification des blocs source par un réseau de Kohonen Multi-dictionnaires.
4. Codage fractal des blocs destination de la partition R, en utilisant les blocs source quantifiés dans les cartes de Kohonen.

Les blocs homogènes sont éliminés de la recherche, un bloc de ce type est assimilé à un bloc constant égale à la moyenne des couleurs des pixels qui le composent.

### 5. Résultats expérimentaux

Cette section présente des résultats expérimentaux montrant l'efficacité de la méthode proposée. Nous avons testé notre algorithme sur une panoplie d'images couleurs.

Nous présentons ici des tests effectués sur deux images de taille 256×256 présentés dans la figures 1 : image de nuages et image de légumes.



FIG. 1 : Les deux images originales

Dans la figures2., les images reconstruites sont présentées, ainsi que les images différences dans la figure 3. Les blocs source utilisés sont de taille 4×4 pixels, les blocs destination sont de taille 2×2. Les cartes utilisées lors du codage sont des cartes contenant 256 neurones où chaque neurone est un bloc de taille égale à la taille des blocs source (4×4 pixels).



FIG. 2 : Les deux images reconstruites



FIG. 3 : Les deux images différence

Il est évident que la qualité de reconstruction dépend fortement du contenu des blocs source quantifiés, nous remarquons dans les images différences des erreurs de codage situées essentiellement aux contours des images.

Mais en général, la qualité des images reconstruites est très bonne et le temps de compression (évalué en secondes) est assez réduit.

**TAB 1 :** Résultats de compression avec une carte de taille 8×8. (Tailles des blocs source =4×4, Taille des Blocs Destination=2×2)

Image	T	V <sub>c</sub>	EQM	PSNR (db)	TC(%)	Durée(s)
Légumes	0,3	3	51,70	30,99	87,17	12
Nuages	0,3	3	27,67	33,70	87,67	34

Les résultats présentés dans la tables 1, où on a utilisé des cartes de taille 16×16, soulignent la rapidité de la phase de codage de l'algorithme proposé. (T : Taux d'apprentissage, V<sub>c</sub> : voisinage, Nombre d'itérations dans la phase d'apprentissage égale à 5). Cela revient essentiellement à la rapidité de construction des dictionnaires avec un apprentissage compétitif de l'algorithme de Kohonen, où les relations topologiques (propriétés topologiques) permettent de construire un système robuste de quantification vectorielle.

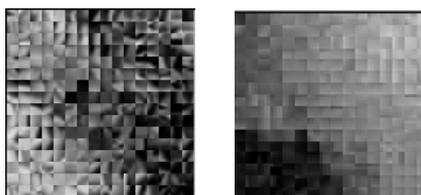


FIG 4 : Codebook des blocs contours à gauche et Codebook des blocs texturés à droite.

Les deux dictionnaires contenant des blocs source quantifiés de taille 4×4 pixels, utilisés lors du codage fractale, sont présentés dans la figure 4 (Cartes de 256 neurones, T=0.3, V=3, Nbr-Iter=5). Remarquons que les neurones voisins sont proches les uns des autres, grâce aux propriétés topologique de la carte de Kohonen.

## 6. Conclusion

Dans ce papier, nous avons présenté une approche destinée à accélérer la phase de codage fractal où La recherche exhaustive pour l'appariement des blocs au niveau de l'ensemble  $D$  (blocs source) a été remplacée par une recherche dans des cartes topologiques de Kohonen.

Les tests ont montré que cette approche a sensiblement diminué le temps de calcul du codeur fractal, tout en maintenant une qualité satisfaisante de l'image reconstruite à des taux de compression importants.

Des systèmes de codage hybride incluant des techniques d'analyse, de pré ou post-traitement d'images, sont de plus en plus courants. Dans ce contexte, l'introduction de mécanismes prenant en compte des propriétés de la perception visuelle est un enjeu important qui se heurte souvent à l'absence d'un modèle global de la vision.

## Références

- [1] A. Skodras, C. Christopoulos, and T. Ebrahimi. *The jpeg 2000 still image compression standard*. *Signal Processing Magazine*, Vol.18 pp.36-58, September 2001.
- [2] A. E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, 1(1) :18–30 (1992).
- [3] Fisher Y., *Fractal Image compression: theory and application*, Springer-Verlag, (1995).
- [4] D.Saupe, M.Ruhl, Evolutionary fractal image compression, Proc. Int. Conf. on Image Processing, ICIP'96, Lausanne(1996).
- [5] R.Hamzaoui and D. Saupe. Combining Fractal Image Compression and Vector Quantization. *IEEE Trans. on Image Processing*, 9(2), pp.197-208, (2000).
- [6] M. Hassaballah , M.M. Makky and Youssef B. Mahdy, *A Fast Fractal Image Compression Method Based Entropy*, Mathematics Department, Faculty of Science, South Valley University, Qena, Egypt , 2004
- [7] C.S. Tong, and M. Pi, Fast Fractal Image Encoding Based on Adaptive Search, *IEEE Transactions on Image processing*, Vol. 10, No. 9, 2001.
- [8]. M. Polvere and M. Nappi. Speed-Up in Fractal Image Coding: Comparison of Methods. *IEEE Trans. on Image Processing*, Vol. 9, No. 6, pp.1002-1009, June 2000.
- [9]. C.S. Tong and W. Man. Adaptive Approximation Nearest Neighbor Search for Fractal Image Compression. *IEEE Trans. on Image Processing*, 11(6), pp.605-615, 2002.
- [10] N.Benamrane,Z.Benahmed Daho, J.Shen, « Medical Images Compression by Neural Networks », Proceeding 3rd International Symposium on image and signal processing and analysis, pp1330-1012, Rome, Italy, September 18-20, 2003.
- [11] D.M. Monro and F. Dudbridge. Approximation of Image Blocks. in Proc. Int. Conf. Acoustics, Speed, Signal Processing, Vol. 3, pp.4585-4588, 1992.