

Optimisation de la compression fractale D'images basée sur les réseaux de neurones

D' BOUKELIF Aoued

Communication Networks ,Architectures
and Mutimedia laboratory
University of S.B.A
aoued@hotmail.com

Mokhtari Mohamed

Université de Sidi Djillali
Faculté des sciences de l'ingénieur
Département de l'informatique
Laboratoire EEDIS

I. Compression fractale

La compression fractale consiste en la détection des auto-similarités présentes au sein d'une image, pour avoir enfin l'IFS qui peut la générer. L'IFS est un ensemble de transformations massiques et spatiales, entre des parties de l'image. Jacquin était le premier à proposer une méthode pour la compression fractale, basée sur le principe de recherche les blocs similaires selon une distance, entre l'image initiale (qui sera l'image destination). Ses blocs de taille $B \times B$ sont considérés comme *blocs destinations*, et l'image des *blocs domaines* de tailles $2B \times 2B$. On dira aussi comparer entre l'image de haute résolution et l'image de basse résolution.

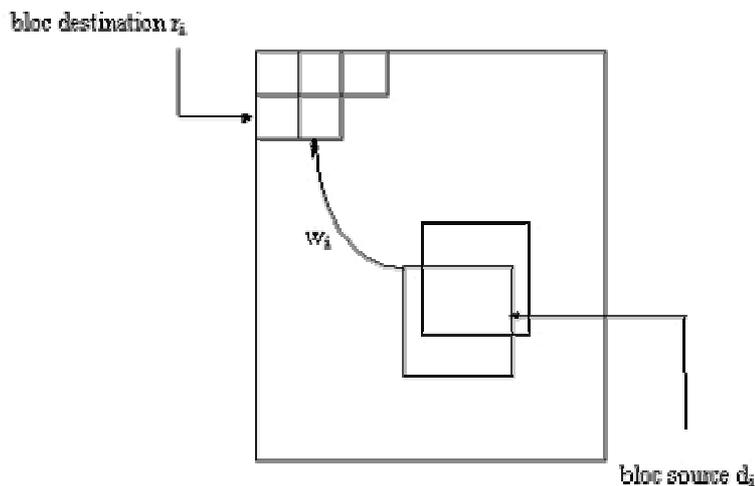


Fig.1 Illustration de la comparaison entre les blocs source r_i et blocs domaine d_i

D'autres transformations sont considérées ; ce sont les *huit isométries*.

Pour chaque bloc source c_i , tous les blocs domaine seront balayés, pour trouver le plus proche d_j , ainsi des paramètres de transformation spatiale a et massique o telle que:

$$c_i = a \cdot d_j + o.$$

Cette façon de partitionner l'image en blocs carrés, de même taille est dite aussi *partition régulière* où le partitionnement est dit carré .prouvais des inconvénients, du fait que le nombre de blocs étant fixe par rapport aux dimensions de l'image, et le partitionnement ne s'adapte pas aux détails de l'image .D'autres méthodes de partitionnement ont été proposées pour adapter les blocs aux détails d'image ;on cite les méthodes de partitionnement suivantes :

- Partitionnement quadtree ,
- Partitionnement rectangulaire .
- Partitionnement HV (Horizontal Vertical)
- Partitionnement à base de polygones .
- Partitionnement triangulaire .

II. Le Problème Inverse

La génération d'une image à partir d'un IFS donnée étant une chose très facile, l'imag étant complètement auto-similaire Son idée est intéressante dès que l'information nécessaire pour construire cette image se réduit aux transformations contractantes Donc avec de très peu d'informations on peut construire une image entière, mais comment définir l'IFS qui génère une image quelconque? Comment peut-on exploiter cette technique à part le côté artistique ? C'est le problème inverse, un problème qui est résoluble selon le théorème de Barnsley, mais très difficile .

Les IFS ne sont efficaces que pour des images typiquement fractales, où la similarité est à tous les niveaux, alors que les images naturelles ne sont pas tout à fait fractales, avec des similarités présentes par régions, d'où la nécessité du recours aux PIFS (Partitioned IFS) .

III. Le Réseau De Neurones Artificiels

Un réseau de neurones artificiel est un ensemble d'unités de calcul, ressemblant aux neurones biologiques, dans la connaissance par l'intermédiaire de l'apprentissage . Les connexions neuronales connues sous le nom de *poids synaptiques* vont stocker la connaissance.

Un réseau de neurones est un ensemble de cellules ou neurones connectés entre elles ; chaque cellule possède une entrée par laquelle elle reçoit une information par une autre cellule, et une sortie à travers laquelle elle envoie une information vers une autre cellule .Un réseau de neurones dispose aussi d'une *fonction d'activation*.

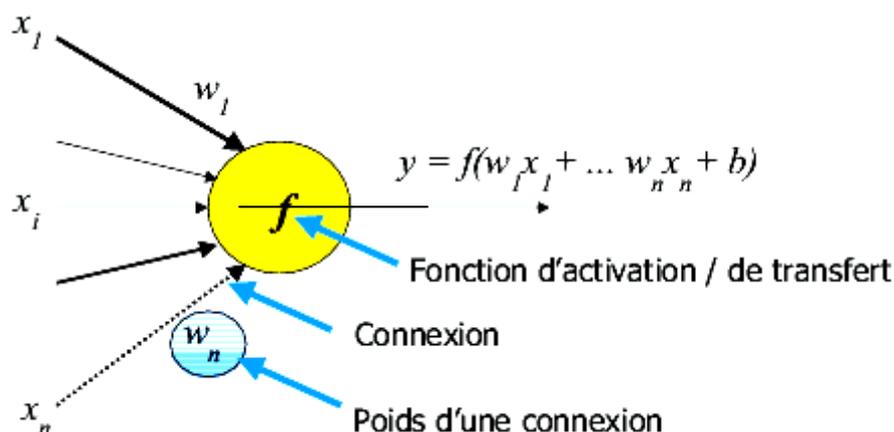


Fig. 2. Un exemple d'un neurone et des connexions.

Il est important de choisir le type de fonction d'activation. On cite parmi les fonctions d'activation :

- Le sigmoïd standard, (ou fonction logistique) : $Y = F(X) = 1/(1 + \exp(-d*X))$.
- La tangente hyperbolique : $Y = 2 / (1 + \exp(-2 * X)) - 1$.
- La fonction Gaussienne : $Y = \exp(-(x^2)/2)$.
- Une fonction à seuil : $Y = 0$ si $X < 0$ et $Y = 1$ si $X > 0$

On ne cesse d'inventer de nouveaux types de réseaux de neurones, toujours mieux adaptés à la recherche de solutions à des problèmes particuliers parmi lesquels :

- Les Perceptrons Multicouches.
- Les Réseaux de Kohonen.
- Les Réseaux de Hopfield.

III.1. Réseaux Kohonen

L'analyse Kohonen est une méthode de classification non supervisée.

Elle a les *propriétés* suivantes:

- Elle représente les données en conservant la topologie.
- Des données proches (dans l'espace d'entrée) vont avoir des représentations proches dans l'espace de sortie et vont donc être classées dans une même classe ou dans des classes voisines.

Un réseau de Kohonen est constitué:

- d'une couche d'entrée:
Tout individu à classer est représenté par un vecteur multidimensionnel (le vecteur d'entrée). A chaque individu est affecté un neurone qui représente le centre de la classe.
- d'une couche de sortie (ou couche de compétition).
Les neurones de cette couche entrent en compétition. Seuls les meilleurs gagnent ("WTA ou Winner Takes All")

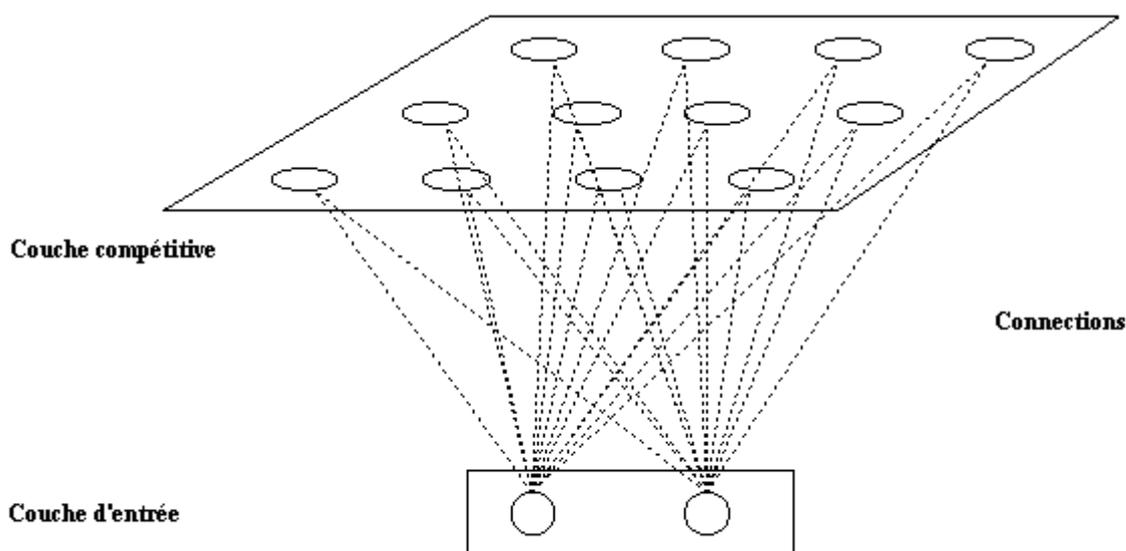


Fig. 3. Réseau de Kohonen

A chaque neurone, on fait correspondre un espace de sortie qui ,en général, est un espace 2D défini par une grille soit rectangulaire soit hexagonale.

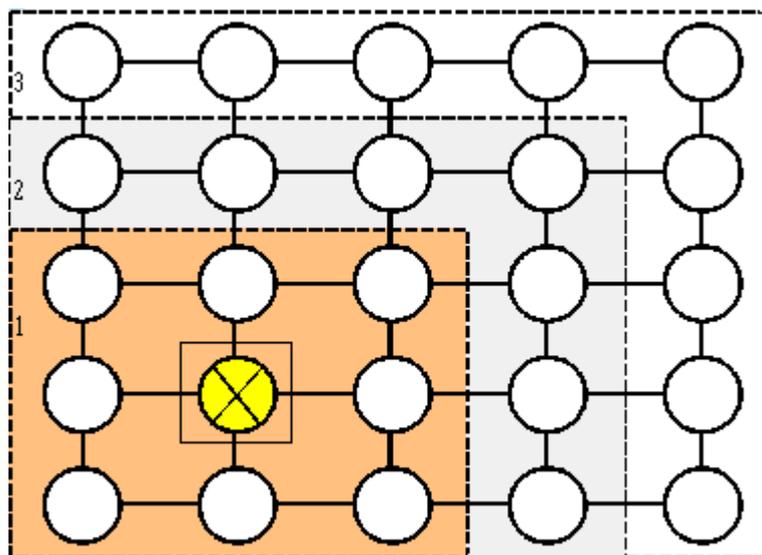


Fig.4 .Grille Rectangulaire

Chaque neurone possède 8 plus proches voisins .

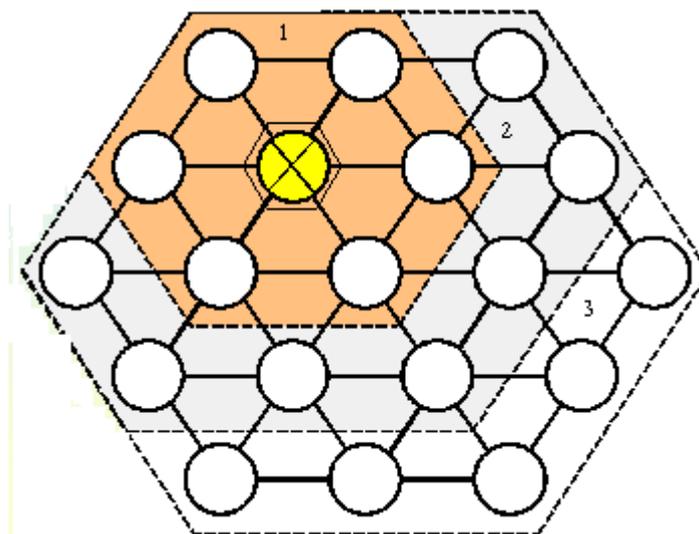
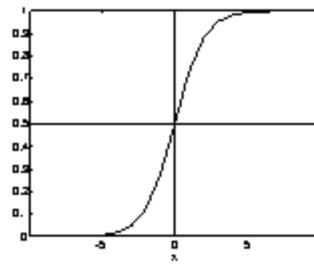


Fig.5. Grille Hexagonale

Chaque neurone possède 6 voisins

III.2. Le Neurone Biologique

Le neurone de Kohonen a une fonction d'activation qui ne prend que des valeurs positives.



$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Fig. 6 .Fonction d'activation d'un neurone de Kohonen

Compétition

1. Fonction à la forme d'un chapeau mexicain :

Chaque neurone reçoit les signaux de ses voisins. Son excitation (ou son inhibition) dépend de la distance et peut être représentée par la fonction ci-après.

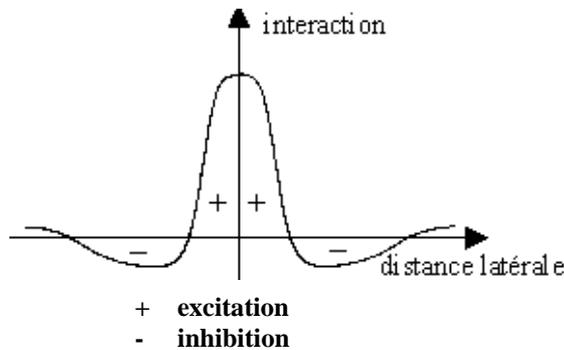


Fig. 6.Fonction à la forme du chapeau mexicain

Les neurones voisins ont une action d'excitation.

Les neurones éloignés ont une action d'inhibition.

La loi d'évolution de la couche fait que le réseau s'organise de telle façon à créer un amas de neurones autour du neurone le plus stimulé par le signal d'entrée. Les autres neurones se stabilisent dans un état d'activation faible.

2. Fonction gaussienne :

Chaque neurone dans le voisinage du neurone gagnant essaie de s'approcher de l'entrée selon leur distance relative . Le neurone gagnant aura la plus grande amplitude, et plus en plus qu'on s'en éloigne l'amplitude diminue jusqu'à ce que ce qu'elle s'annule.

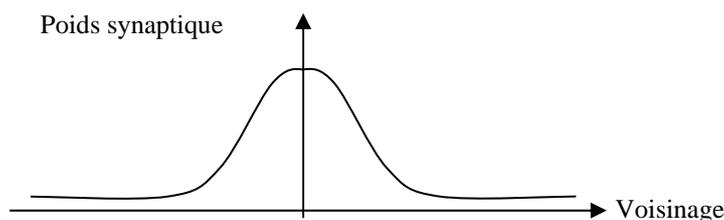


Fig. 7. Fonction gaussienne

III.3. Algorithme

1. Prétraitement des données

Chaque vecteur d'entrée V est normalisé de telle façon à ce que sa longueur soit égale à 1, ainsi que les poids initiaux W (générés aléatoirement) des neurones de compétition .

2. Présentation des données

Les données d'entrée peuvent être présentées soit dans leur ordre initial soit de façon aléatoire.

3. Recherche du neurone gagnant

La recherche le neurone dit "gagnant" se fait en minimisant $(V-W)$.

$$\|w_r - v\| = \min_r \|w_r - v\|$$

Les neurones voisins du gagnant sont modifiés, à chaque itération, comme suit:

$W=W+\alpha(V-W)$, α étant la vitesse d'apprentissage. Elle peut être linéaire et décroissante en fonction du temps ou gaussienne des distances et décroissante en fonction du temps.

4. Visualisation de la couche de sortie

La visualisation de la carte finale est en général faite en projetant les neurones gagnants dans un espace 3D/2D ("Non linear mapping").

IV. Les Réseaux de Neurones dans la compression fractale

Le grand problème rencontré dans la compression, est le temps de calcul nécessaire pour trouver les paramètres des transformations contractantes, dû au grand nombre de blocs à comparer pour trouver ceux qui se ressemblent.

Le type de réseau de neurones qui a été proposé pour être utilisé dans la compression fractale est le SOM . L'idée est d'utiliser la capacité des réseaux SOM dans la classification, dans la réduction de l'espace de recherche des blocs domaines ; ainsi on réduit le temps consommé pour la recherche.

Le processus se résume ainsi :

- Apprentissage des blocs domaines, et identification des neurones correspondant à chacun d'eux.
- Reconnaissance des blocs sources, et identifications de leurs neurones correspondants.
- Comparaison des blocs sources avec les blocs domaines associés aux neurones qui leurs correspondent, et ceux dans le voisinage.

IV.1. Les Algorithmes proposés par A.Bogdan et H.Meadows [2]

-Algorithme d'apprentissage :

L'apprentissage est de type compétitif , et le réseau de neurones consiste en une couche d'entrée et une couche de sortie arrangée en une matrice rectangulaire de neurones. Un neurone N_{mn} est un simple processeur ayant une matrice de poids w_{mn} ayant la taille du bloc de partition b_{ij} .On applique le bloc b_{ij} à chaque neurone en entrée. Chaque neurone en sortie aura en sortie :

$$u_{mn} = d(w_{m,n}, b_{ij})$$

d est une mesure de similarité.

On applique l'apprentissage sur un ensemble de blocs b_{ij} qui sont les blocs domaine, d'une image test ou d'un ensemble d'images représentatives.

- Algorithme proposé par A.Bogdan, H.E.Meadows

- 1- Initialiser les poids $w_{m,n}$ de chaque neurone N_{mn} avec des valeurs aléatoires, ou utiliser une carte prédéfinie obtenue avec un autre apprentissage.
- 2- Un block b_{ij} est présenté au réseau et chaque neurone en sortie calculera la distorsion u_{mn} .
- 3- Sélection du neurone N_{max} qui stocke dans les poids le modèle le plus proche à l'entrée b_{ij} .
- 4- Mise à jour des poids du neurone sélectionné et de ces voisins.
- 5- Répéter les étapes 2) et 3) pour tous les vecteurs.

- Algorithme du codage fractal utilisant les carte auto- organisatrice :

La comparaison sera faite entre les blocs source de l'image originale et les blocs domaines de la même image réduite d'un facteur de deux dans chaque direction appelée D. Les blocs domaines seront l'ensemble d'apprentissage. L'algorithme du codage fractal avec le réseau de neurone, proposé par A.Bogdan et H.Meadows dans [2] :

- 1- Construire la bibliothèque dans le réseau SOM utilisant les blocs de l'image D en tant que ensemble d'apprentissage de la façon suivante :
 - 1.a - Charger le réseau SOM avec la librairie initiale ou assigner des poids aléatoire pour tous le neurones.
 - 1.b - Balayer l'image D avec une fenêtre de taille (b_x, b_y) et de pas d'un pixel .Chaque bloc sera passé en tant qu' individu à apprendre par le réseau .
 - 1.c - Répéter le balayage de D n fois ou jusqu'à ce que la carte converge (état d'équilibre).
- 2- Etablir une correspondance entre chaque bloc de D et chaque neurone. Chaque neurone garde une liste de quelques pixels de D, proche selon la distance de similarité d au modèle enregistré dans les poids.
- 3- Faire la partition de l'image originale en bloc source de la même taille que les précédents .Etablir la correspondance entre chaque bloc b_{ij} et neurone, en présentant ce bloc au réseau de neurones en sélectionnant le neurone N_{kl} tel que $kl = \arg(\min_{mn} d(w_{mn}, b_{ij}))$.
- 4- Etablir une correspondance entre chaque partition b_{ij} et un bloc arbitraire de D utilisant le réseau SOM.
 - 4.a - Explorer la listes des blocs domaines associés au neurone N_{kl} et son voisinage, c'est à dire les neurones $N_{k+/-p, l+/-q}$ ($p, q = 0, 1$) , et trouver le bloc d_{sr} qui est le plus proche.
 - 4.b - Répéter a) pour tous les blocs b_{ij} .

IV.2. Methode proposee par welstead [3]:

Une autre méthode utilisant ce même type de réseau de neurones, mais d'une façon améliorée, c'est de coder les blocs non pas sur la base des pixels, mais sur la base des caractères . Stephen Welstead a mentionné 6 caractères, qui se calculeront sur les blocs .

1. *Déviatiion standard.*
2. *Obliquité* : somme des cubes des différences entre les valeurs des pixels et la moyenne des cellules normalisées par la déviatiion standard.
3. *Contraste voisin* : différence moyenne entre les valeurs des pixels adjacents.
4. *Bêta* : mesure combien les valeurs des pixels changent par rapport à la valeur de la valeur du centre de la cellule.
5. *Gradient horizontal.*
6. *Gradient vertical.*

V. Implémentatiion :

Les deux méthodes ont été développées dans un environnement Matlab sur la plate forme Windows, sur une machine Pentium 4. La possibilité d'utiliser les réseaux de Kohonen dans la compression fractale a été démontrée.

Résultats de la simulation

Première méthode :

La carte utilisée est de dimension 40×40, et le vecteur d'entrée et de longueur de 16. On balaye l'image à basse résolution **D** qui est l'ensemble des blocs destination de taille 4×4, avec un pas d'un pixel .Le nombre de ces blocs est 15625, et le nombre de blocs source de même taille est 4096. Les blocs devront être normalisés avant d'être présenté au réseau. Ce tableau présente les résultats de compression après chaque itération pour l'image Cameraman.

Itération	1	2	3	4	5	6	7
PSNR dB	16.6260	28.7657	28.6613	28.5714	28.5657	28.2667	28.7776

Pour l'image du cameraman, l'apprentissage peut s'arrêter à l'itération 7, et pour l'image maison à l'itération 4 .Voici les image restituées :



Fig. 8. L'image du Cameraman : A gauche l'image originale, et à droite l'image restituée au PSNR de 28.7772



Fig. 9. L'image de Maison : A gauche l'image originale, et à droite l'image restituée au PSNR de 29.3895

NOTRE METHODE : Algorithme adaptatif et réseau de Kohonen

Le réseau de Kohonen a été appliqué avec une compression fractale utilisant *l'algorithme adaptatif*. L'algorithme adaptatif consiste à chercher pour un bloc source le meilleur bloc destination avec les paramètres de transformation contractante équivalente. Si la distorsion entre le bloc source, et le bloc destination transformé est supérieure à un seuil déterminé, alors on subdivise le bloc source en quatre sous blocs de tailles égales, et on répète le même procédé aux petits blocs, jusqu'à ce que la distorsion soit inférieure au seuil, ou que la taille minimale soit atteinte. Autrement dit pour tout bloc source r_i , on essaie de trouver le bloc destination d_i , avec la transformation affine w_i , ayant le paramètre de contraste s_i , et le paramètre de luminosité o_i , tel que $w_i(d_i)$ soit le plus proche au bloc r_i , ce qui revient à ce que la distance $d(w_i(d_i), r_i)$ soit minimale. Si cette quantité est inférieure au seuil, alors subdiviser r_i en quatre sous blocs et répéter l'opération sur ces blocs.

Le tableau suivant résume les résultats obtenus, le seuil est fixé à $\sqrt{5}$, et la taille minimale autorisée est 4×4 . On remarque que le temps a été réduit à un tiers de celui consommé dans la méthode ordinaire. Le champ de recherche était réduit avec le réseau de neurone, ce qui a engendré une réduction du temps de codage.

	Nombre de blocs source	Temps	PSNR	Ratio
Cameraman(1)	2452	21 mn 8 sec	29.389	5.34 :1
Cameraman	2530	7 mn 56 sec	27.418	5.18 :1
Maison	3478	11 mn 25 sec	22.585	3.77 :1

Voici les images restituées :



Fig. 10. Les images restituées avec l'algorithme adaptatif utilisant les réseaux de Kohonen .

Conclusion

Dans ce papier on a présenté une nouvelle façon d'exploiter la technique de compression fractale d'images de façon à optimiser le temps de codage, qui est très considérablement long.

Une autre piste à investiguer consiste à classifier les blocs sur la base *sensitivité du système visuel humain*, et avec des cartes topologiques multidimensionnelles.

Bibliographie

- [1] Compression des images fixes par fractales basée sur la triangulation de Delaunay et la quantification vectorielle. Kaddour Chakib, Aissa Brahim Salim.
Université des sciences et de la technologie Houari Boumediene institut d'informatique Algérie
- [2] Kohonen neural network for image coding based on iteration transformation theory.
A.Bogdan, H.E.Meadows. Columbia University .
- [3] Self Organizing Neural Network Domain For Fractal Image Coding. Stephen Welstead.
Colsa Corporation. International conference artificial conference intelligence and soft computing July 27-31 1997 .
- [4] Cours de Neuro-mimétique .Michael Aupetit , Février 1999.
- [5] Réseaux de neurons .Marc Olivier La Barre , Mai 2002.
- [6] Kohonen Neural network for image coding based on iteration transformation theory.A.
Bogdan, HE Meadows.Colombia University.
- [7] Self organizing Neural network domain for fractal image coding. Stephen Welstead. Colsa Corporation . International artificial intelligence and soft computing conference.July 27-31, 2003.
- [8] Réseaux de neurones .Formation avancée : <http://www.pmsi.fr/neurini2.htm>
- [9] Carte auto-organisatrice de Kohonen : <http://www.scico.u-Bordeaux2.fr/~corsini/pedagogie/ANN/Main>
- [10] Image coding based on fractal theory of iterated Contractive image transformations.
Venu-Gopal Puram .December 2003.