

COMPRESSION FRACTALE

J.-L. Dugelay

Institut **EURECOM**, département Communications MultiMédia
2229, route des Crêtes, B.P. 193, 06904 Sophia Antipolis Cedex
Tél. : +33 - 93 00 26 41 _ e-mail : dugelay@eurecom.fr _ Fax : +33 - 93 00 26 27

1. INTRODUCTION

La notion d'I.F.S. "Iterated Function Systems" a été inventée par le mathématicien J. Hutchinson au début des années 80. Elle désigne l'ensemble des processus itératifs qui possèdent la propriété remarquable de converger vers un élément fixe indépendamment de leur initialisation. Le point fixe est appelé "l'attracteur" de l'I.F.S. Cette notion s'inscrit dans le cadre d'une théorie plus générale due au mathématicien d'I.B.M. B.Mandelbrot qui porte le nom de géométrie fractale. Cette nouvelle géométrie que l'on nomme parfois géométrie de la nature constitue une alternative au monde Euclidien pour la description de la complexité des objets naturels. Hutchinson a démontré que l'attracteur d'un I.F.S. est de type fractal, et a ainsi proposé un algorithme générateur d'images fractales.

2. LA NOTION D'IFS

Par un processus itératif, utilisant des transformations géométriques affines contractives, les I.F.S. permettent de générer des images complexes. Toute l'information nécessaire à la formation de l'image se trouve encodée dans quelques règles simples. A titre d'exemple, l'image intitulée "la fougère de Barnsley" a été construite à partir des quatre transformations affines indiquées ci-dessous.

$$\omega_1 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0,85 & 0,04 \\ -0,04 & 0,85 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0,02 \\ 0,08 \end{bmatrix}$$

$$\omega_2 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -0,13 & 0,24 \\ 0,22 & 0,20 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0,12 \\ -0,27 \end{bmatrix}$$

$$\omega_3 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0,18 & -0,24 \\ 0,21 & 0,20 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -0,12 \\ -0,30 \end{bmatrix}$$

$$\omega_4 \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0,16 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ -0,42 \end{bmatrix}$$

Code IFS



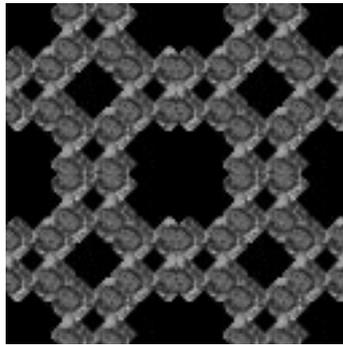
la fougère de Barnsley

Pour préciser davantage la notion d'I.F.S. en compression d'images, considérons maintenant une photocopieuse d'un type un peu particulier : le passage d'une image à la suivante s'effectue par rotation de 45° puis réduction d'un facteur 1/2 et reproduction en 4 exemplaires. Les 4 imageries obtenues sont alors disposées au centre de 4 carrés identiques résultant du partitionnement d'une image uniforme noire. Les parties des imageries se trouvant hors du carré sur lequel elles sont disposées sont supprimées.

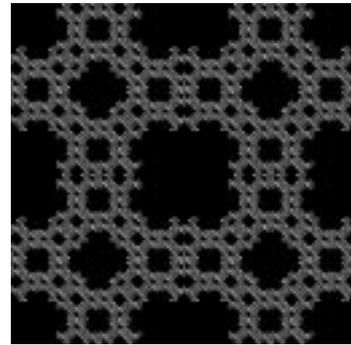
L'illustration suivante montre l'utilisation en boucle¹ d'une telle photocopieuse, à partir de deux images distinctes "x" et "y".



Une Image initiale "x"



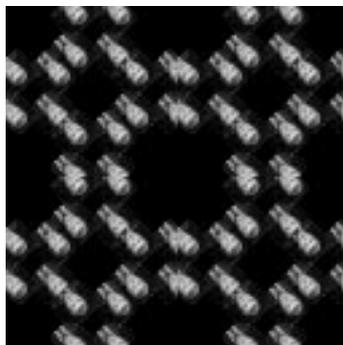
3ème copie



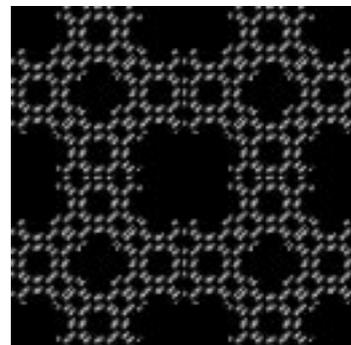
5ième copie



Une image initiale "y"



3ème copie



5ième copie

Nous constatons qu'après quelques itérations de notre processus, nous obtenons le même motif final quelle que soit la nature du document d'entrée. L'image résultant² du processus n'est donc pas fonction de l'image entrante mais des caractéristiques de la machine.

En compression, nous chercherons à résoudre le problème inverse: étant donnée une image, déterminer les caractéristiques de la photocopieuse permettant de la générer à partir d'une image initiale quelconque. Le problème essentiel est d'adapter ce principe afin de déterminer un ensemble de transformations représentant un volume d'informations assez faible et permettant néanmoins d'obtenir une image codée-décodée la plus proche possible, au sens d'une distance, de l'image originale. L'algorithme de référence est l'algorithme proposé par A. Jacquin [1]. Cet algorithme introduit la notion de local-IFS.

3. L'ALGORITHME DE A. JACQUIN

3.1 CODAGE

L'étape de codage consiste à identifier les similarités qui existent dans l'image. Pour ce faire, une partition de l'image est réalisée, et ce à deux niveaux, typiquement en blocs carrés de taille $B \times B$ "range" et $2B \times 2B$ "domain". Il s'agit ensuite d'apparier les blocs

¹le document de sortie constitue le document d'entrée de la prochaine photocopie.

²cette illustration est limitée à l'aspect géométrique et n'inclut pas l'aspect photométrique.

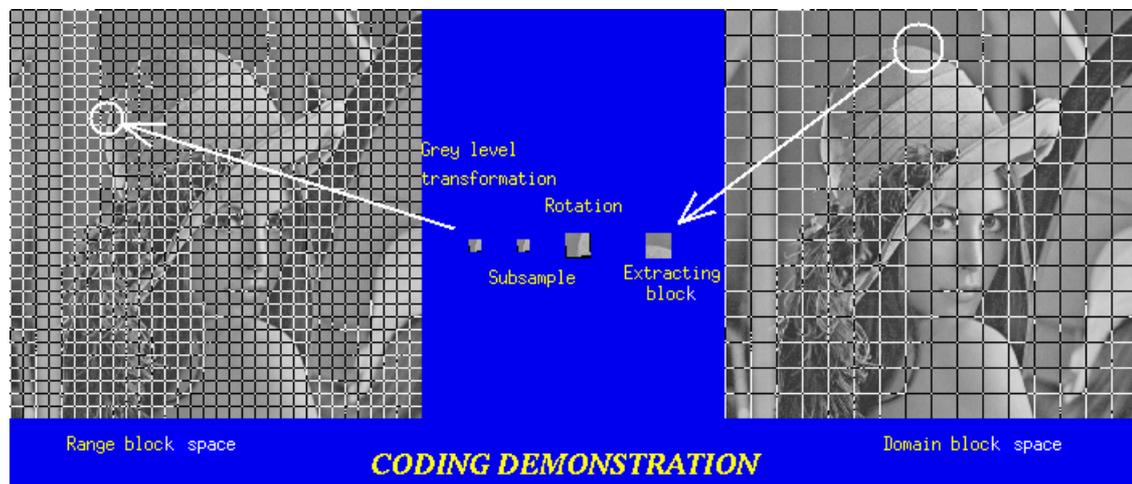
entre ces deux niveaux sur critère d'erreur quadratique moyenne en luminance³ (voir illustration "coding demonstration") :

$$\text{Erreur} = \sum_{(i,j) \in \text{bloc}} \{W_r(D)(i,j) - R(i,j)\}^2$$

avec $R(i,j)$, le niveau de gris associé au "pixel" (i,j) du "range block";
 $W_r(D)$, la transformation affine appliquée au "domain block".

Avant appariement, les "domain blocks" sont transformés comme suit:

- un sous-échantillonnage par un facteur 2 (en horizontal et en vertical) obtenu par moyennage;
- une transformation géométrique, ou plus exactement une isométrie choisie dans un dictionnaire prédéfini;
- une transformation photométrique ("shift" et "scale")⁴.



3.2 DECODAGE

Pour décoder l'image à partir de son code I.F.S. et d'une image initiale quelconque, l'algorithme procède comme suit :

L'image initiale (notée x_0) est partitionnée en blocs carrés. Chaque bloc de l'image est calculé à partir du "domain block" associé dans l'image elle-même, en lui appliquant la transformation affine contractive définie lors de l'étape de codage. Ainsi, l'image x_1 est obtenue. L'algorithme de décodage itère ce processus pour obtenir l'image x_2 à partir de x_1, \dots , jusqu'à obtenir l'image codée-décodée (voir illustration "decoding demonstration", page suivante).

³il existe une analogie avec la technique de "block-matching" en estimation de mouvement, sauf qu'ici on opère dans l'image elle-même et on effectue une transformation géométrique et photométrique, à savoir: un sous-échantillonnage du bloc, une isométrie, un "shift" et un "scale" sur les valeurs de luminance.

⁴dans le cadre de cette étude, ces deux paramètres sont identifiés par moindre carrés [2].

