Texture analysis using fractal probability functions

Jacques Lévy Véhel

UNITE DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

1992

ème anniversaire

N° 1707

Programme 4
Robotique, Image et Vision

TEXTURE ANALYSIS USING FRACTAL PROBABILITY FUNCTIONS

Jacques LEVY VEHEL

Juin 1992

Texture Analysis using

Fractal Probability Functions

Analyse de Textures par

Fonctions de Probabilités Fractales

Thème 4

Jacques LEVY VEHEL

INRIA, Domaine de Voluceau,Rocquencourt B.P. 105, 78153 Le Chesnay CEDEX, France.

Email: jlv@bora.inria.fr Tel.1-39 63 54 73

Abstract

In this work, we use iterations of contractive functions to build textures. At each iteration, one of the contractions is chosen at random according to place-dependent probabilities. This scheme allows to encode and reconstruct exactly a wide class of textured images, thus providing a compact representation usable for analysis and data compression.

Dans ce travail, nous utilisons des itérations de fonctions contractantes pour construire des textures. A chaque itération, une contractions est choisie au hasard selon des fonctions spatiales de probabilités. Cette méthode permet de coder et de reconstruire exactement une large classe de textures, fournissant une représentation compacte utilisable pour l'analyse et la compression des images.

1 Introduction

The aim of this work is to apply the theory of Iterated Function System (IFS) to the coding of textured images. IFS have been widely studied in the last years, within the framework of Fractal Geometry or not (see for instance [2, 4, 3, 1, 5, 6, 7, 9, 11, 12]). They allow to code and reconstruct complex objects using only a small number of parameters. Applications include data compression, image synthesis and the study of dynamical systems.

We shall first make a brief presentation of the theory, before concentrating on an extension of IFS that uses place-dependent probabilities. We then introduce our model for texture analysis, show some results on natural and synthetised textures and discuss the validity of this approach.

2 General theory of IFS

We briefly recall the main aspect of the theory.

Let K be a d-dimensional Euclidean space.

An IFS is a couple (W, P) where:

- $W = \{w_1, \dots, w_n\}$ is a finite set of affine mappings of K into itself. Each w_i is a strict contraction (all eigenvalues have a magnitude less than one).
- $P = \{p_1, \dots, p_n\}$ is a set of probabilities: $\forall i, 0 \leq p_i \leq 1, \sum p_i = 1$.

We then consider the following process:

- choose as a starting point z_0, the fixed point of one of the w_i.
- randomly choose a map w_i (with probability p_i) and compute $z_1 = w_i(z_0)$.
- repeat this process a great number of times.

Then the set A of all computed points is a compact one, called the attractor of the IFS.

Thus, with every IFS is associated a unique attractor A.

The following theorem holds (cf. [2]):

If (W, P) is an IFS and A its attractor, then:

$$A = \cup_{i=1}^{n} w_i(A) \tag{1}$$

Let us make some remarks about this equality:

- A is a union of continuously altered shrunken copies of itself, which shows that A is a fractal.
- The tiling of A with copies of itself may be a "lazy tiling", i.e. the tiles may overlap each other.
- A does not depend on the $p_i's$, but only on the contractions. What depends on the probabilities is the number of times each point of A is "visited", or, in image analysis terms, the grey levels distribution on the attractor. From a mathematical point of view, it can be shown that there is an unique measure μ associated with (W, P), whose support is A and that verifies :

  $\forall E, A \supset E$:

  $$\mu(E) = \sum_{i=1}^{n} p_i w_i^{-1}(E)$$

  μ is called the p-balanced measure associated with (W, P), and this equation expresses the fact that μ is invariant with respect to (W, P). In addition, it can be shown that μ is also attractive, which means that infinite successive applications of (W, P) to any measure ν on A leads to μ. Thus, two sorts of fairly independent properties are associated with an IFS : set properties and measure properties.

The important "Collage Theorem" holds as follows (cf. [3]):

Let the contraction mappings $w_i : K \to K, i = 1, \dots, n$ be chosen such that:

$$h(L, \cup_{i=1}^{n} w_n(L)) < \varepsilon, \quad \text{for some } \varepsilon > 0$$

then:

$$h(L, A) < \frac{\varepsilon}{1-s}$$

where:

- A is the attractor of the IFS.
- h is the Haussdorff metric:

$$h(B, C) = \max[\max_{x \in B}(\min_{y \in C}(d(x, y))), \max_{y \in C}(\min_{x \in B}(d(x, y)))] \tag{2}$$

- s is the highest norm of the $w_i's$:

$$0 < s < 1$$

  and

$$d(w_i(x), w_i(y)) \leq s d(x, y), \quad \forall x, y \in K, \forall i$$

It follows that, to find a suitable set of maps to reconstruct A, we need only make an approximate covering of "lazy tiling" of L by continuously distorted smaller copies of itself.

The fractal objects generated with this method include natural objects (leaves, fern, clouds, ...), and a large number of other complex objects (Dragon curve, Sierpinsky gasket, Koch island, ...).

Many authors have worked on the resolution of the following inverse problem:

"Given a set A, find some w_i that encode A."

See for instance [4, 15, 12]. Some results are interesting, but all methods have severe drawbacks: large computing time, need of manual control, huge number of contractions necessary.

In this work, however, we are not interesting in coding the shape of an object, but rather its texture, that is the measure of the IFS on its attractor, which depends on both the w_i and the p_i. Our problem can be stated as follows:

"Given a texture T, can we find some (w_i, p_i) such that the associated attractor supports texture T."

We shall proove in this paper that this last inverse problem unexpectingly leads to solutions far easier and of better quality than the solutions to the first inverse problem about shapes.

3 Measure theory for IFS

We first recall an extension to the theory that can be found in [3].

3.1 Notations

$W = \{w_1, \ldots, w_n\}$ is again a finite set of mappings of K (complete metric space) into itself. Each w_i is a strict contraction.

But now $P = \{p_1, \ldots, p_n\}$ is a set of probability functions that verify:

- $\forall i, \forall x \in K, p_i(x) \geq 0$
- $\forall x \in K, \sum_{i=1}^{n} p_i(x) = 1$
- Let:

$$\phi_i(t) = \sup_{d(x,y) \leq t} (\mid p_i(x) - p_i(y) \mid)$$

Then:

$\frac{\phi_i(t)}{t}$ is summable over $[0, \alpha[$, for a certain $\alpha > 0$ (Dini's condition).

- $\forall (x,y) \in K^2$:

$$\sum_{i=1}^{n} p_i(x) d^q(w_i(x), w_i(y)) < r^q d^q(x,y)$$

for $r < 1$ and $q > 0$.

- $\forall (x,y) \in K^2$:

$$\sum_{i/d(w_i(x),w_i(y)) \le rd(x,y)}^{n} p_i(x) p_i(y) \ge \delta$$

for a $\delta > 0$

The attractor of the IFS (W, P) is still denoted by A.

Let M be the set of Borel regular measures on K having bounded support and finite mass.

Let f be a continuous function on K, and suppose that f is Lipschitz. We define:

$$f_\# : \begin{array}{l} M \to M \\ \mu \to f_\# \mu \end{array}$$

$f_\# \mu$ is defined as follows:

For $E, K \supset E$:

$$f_\# \mu(E) = \mu(f^{-1}(E))$$

the support of a measure μ is the closed set:

$$\text{spt } \mu = X \setminus \cup \{V : V \text{ open}, \mu(V) = 0\}$$

3.2 Theorem

We define:

$$W(\gamma) = \sum_{i=1}^{n} p_i w_i \#(\gamma) \quad \text{for } \gamma \in M$$

Then there exists a unique measure $\mu \in M$ such that:

$$W(\mu) = \mu \tag{3}$$

This theorem follows from the fact that W is a contraction map in M for the metric:

$$L(\mu, \gamma) = \sup\{\mu(\phi) - \gamma(\phi), \phi : K \to \mathbf{R}, \phi \text{ Lipschitz}\}.$$

for a detailed proof see [9].

This measure is called the invariant measure with respect to (W, P) and is such that:

$$\text{spt}\mu = A$$

Besides, we have the following properties:

- for any function a: $N^* \to \{1 \ldots n\}$

  $$\cap_{\rho=1}^{\infty} w_{a(1)} \circ w_{a(2)} \ldots \circ w_{a(\rho)}(A)$$

  is a singleton $x_a \in A$ (see[9]). This simply means that if you apply recursively at A a randomly chosen sequence of w_i, you converge to a single point in A.

- If we denote by Π the function:

  $$C(n) \to K$$

  $$a \to x_a,$$

  where:

  $C(n)$ is the set of maps: a: $N^* \to \{1 \ldots n\}$

  then we have:

  $$\mu = \Pi_{\#}\sigma$$

where σ is the product measure on $C(n)$ induced by the measure p_i on each factor $\{1 \ldots n\}$. This means that the probability to obtain each point x lying upon A is proportional to the product of the p_i associated with the sequence of w_{i_p} used to obtain x.

Note that μ generally fails to have a density, hence usual techniques for studying measures are of little help here.

4 Model for Texture Analysis

We now propose an entirely new application of the former theory to image processing.

By invariance of μ, we have, for each $E, E \subset A$:

$$W(\mu)(E) = \mu(E)$$

that is:

$$\sum_{k=1}^{n} \int_{w_k^{-1}(E)} p_k(x)\mu(x) = \mu(E) \tag{4}$$

The basic idea of our work is then to consider an image as being the attractor of an IFS (W, P) and to identify the invariant measure μ with the grey levels distribution of the image. We then find a mean to compute the probability functions p_i that generates μ.

Thus our measure μ is known by its values on the set of all the pixels of the image, at a given resolution. In theory, it is possible to compute $\mu(E)$ for any subset E in an image. However, the resolution being fixed, we don't have any idea of the value of $\mu(E)$ when E is smaller than a pixel. More that that, since usually μ has no density, it is impossible to compute $\mu(E)$ when E is not an exact collection of pixels. By this, we mean that speaking of the measure of half a pixel, or of one pixel and a half, has no sense, at a given resolution. The measure may be so irregular that it is totally unfounded to perform any sub-pixel computation, as is usually done.

Hence, we may write, for a pixel (i,j):

$$\sum_{k=1}^{n} p_k(w_k^{-1}(i,j))\mu(w_k^{-1}(i,j)) = \mu(i,j) \tag{5}$$

but we have in general no way to evaluate $\mu(w_k^{-1}(i,j)$ if we only know our image at a single resolution because most of the times, $w_k^{-1}(i,j)$ will not coincide with a set of pixels.

Making any kind of interpolation between the available data would lead to completely false results, since μ has no density.

To solve this problem, we shall focus on a peculiar case of attractor that allows correct computations.

Let:

$$\begin{aligned} w_1(x,y) &= (0.5x+0.5, 0.5y+0.5) \\ w_2(x,y) &= (0.5x-0.5, 0.5y+0.5) \\ w_3(x,y) &= (0.5x-0.5, 0.5y-0.5) \\ w_4(x,y) &= (0.5x+0.5, 0.5y-0.5) \end{aligned}$$

The attractor is a square C, obtained without overlapping (see figure 1)

Let B be a subset of $w_1(C)$. Then:

$$\begin{aligned} \forall x, x \in w_1^{-1}(B) &\Rightarrow x \in C \\ \forall i \geq 2, x \in w_i^{-1}(B) &\Rightarrow x \notin C \end{aligned}$$

Thus, equation 4 becomes:

$$\int_{w_1^{-1}(B)} p_1(x)\mu(dx) = \mu(B)$$

In order to obtain an explicit expression for p_1, we have to inverse this formula.

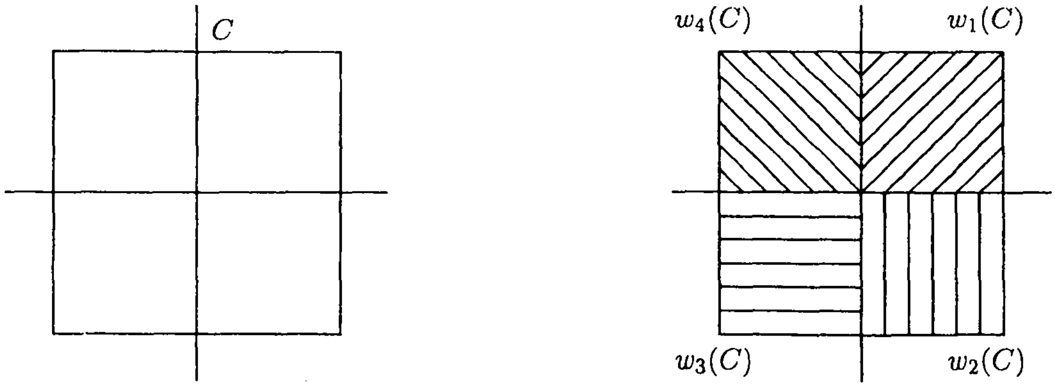

Figure 1: Case of the square

More precisely, what we are going to obtain is an expression of p_1 that will be valid at a given resolution, namely half the initial resolution of the square.

Let K_R be the set of pixels at resolution R, and K_{2R} the set of pixels at resolution $2R$, which we assume to be the resolution at which the texture is given.

We now consider the w_i as functions from K_R to K_{2R}

$$w_i \; : \; K_R \rightarrow K_{2R}$$

$$(I, J) \rightarrow (i, j) = w_i(I, J)$$

where (I, J) is the R-resolution pixel centered on (I, J) and (i, j) the $2R$-resolution pixel centered on (i, j) (see figure 2).

Each w_i is then injective from C_R to C_{2R} where C_R is the R-resolution square.

Moreover, each point of C_{2R} has a unique inverse image in C_R and hence is reached by one and only one w_i.

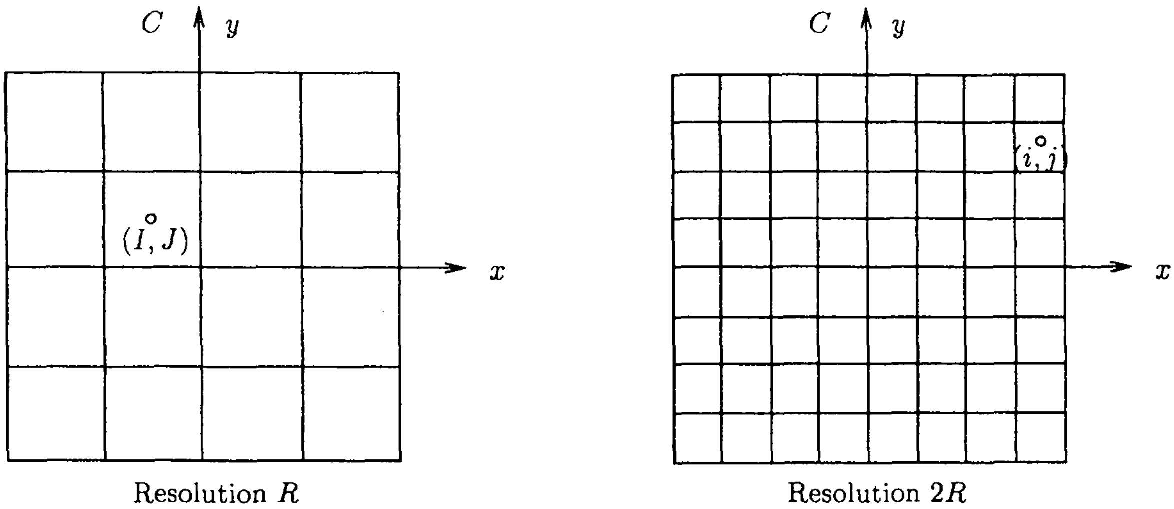

Figure 2: Changement of the resolution for the square

For a discrete process we may write:

$$\mu(x_l) = \sum_{k=1}^{n} \mu(w_k^{-1}(x_l))p_k(w_k^{-1}(x_l))$$

We take for x_l a pixel of C_{2R}, whose inverse image will be a pixel of C_R:

$$\mu(i,j) = \sum_{k=1}^{n} \mu(w_k^{-1}(i,j))p_k(w_k^{-1}(i,j))$$

Let (i,j) be in the first quadrant of the plane; its inverse images by w_2, w_3, w_4 are outside the square C_R, hence their measure is zero:

$$\mu(w_2^{-1}(i,j)) = \mu(w_3^{-1}(i,j)) = \mu(w_4^{-1}(i,j)) = 0$$

Let $(I,J) = w_1^{-1}(i,j)$.

Then:

$$\mu(i,j) = \mu(I,J)p_1(I,J)$$

Since (I,J) is inside C_R, its measure is non zero:

$$p_1(I,J) = \frac{\mu(i,j)}{\mu(I,J)}$$

This is the equation we shall use. It holds for any (I, J) inside C_R.

In order to recall that $\mu(i,j)$ must be measured on the $2R$-resolution grid and $\mu(I, J)$ on the R-resolution grid, we write:

$$p_k^R(I,J) = \frac{\mu_{2R}(w_k(I,J))}{\mu_R(I,J)} \tag{6}$$

for $k = 1 \ldots 4$.

These formulae give the values of the four probability functions that must be used to construct a given measure μ. As has already been said, the p_k are only known at half the initial resolution.

4.1 Results

To verify numerically our theory and show the power of the method, we first generate textures with different probability functions.

In order to apply our formula, we must compute the texture C_R, then C_{2R}, then obtain the expression of the p_k.

We do not show the comparison between theoritical and computed p_i on images, because they are strictly identical: the results fit perfectly.

a) Polynomial probability functions:

We set:

$$p_k^R(I,J) = \frac{\sum_{n=0}^{N}\sum_{m=0}^{M} a_{m,n}^k I^M J^N}{\sum_{i=1}^{4} p_i^R(I,J)}$$

with $a_{m,n} \geq 0$ for each m, n and $a_{00} > 0$.

Figure 3 shows the resulting texture along with p_1 for the following values:

$$
\begin{aligned}
p_1 &= 0.05x^2 + 0.6y^2 + 1.5 \\
p_2 &= 0.5x^2 + y^2 + 0.1 \\
p_3 &= 0.8x^2 + 0.5y^2 + 0.03 \\
p_4 &= 0.7x^2 + 0.3y^2 + 0.9
\end{aligned}
$$

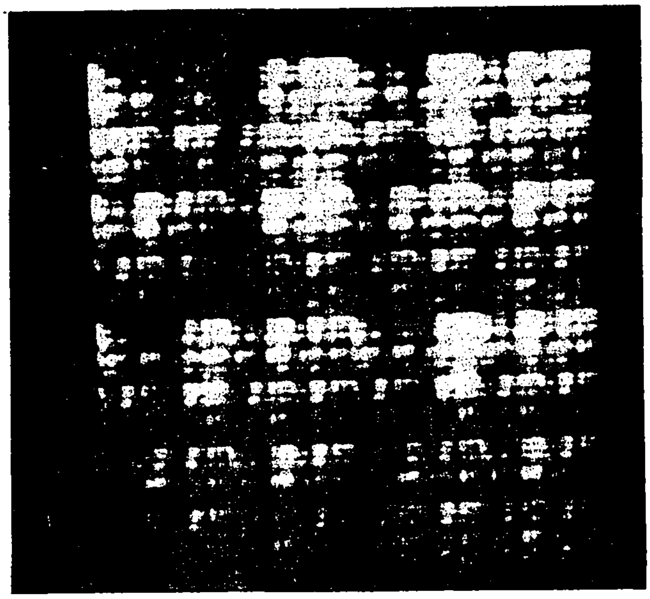

Figure 3: texture obtained with polynomials probabilities

b) Modulo probability functions:

We take:

$$p_k^R(I,J) = \frac{(a_k \bmod(I, I_0) + b_k \bmod(J, J_0) + \alpha_k)}{\sum_{i=1}^{n} p_i^R(I,J)}$$

where $a_k, b_k \geq 0; \alpha_k > 0; I_0, J_0 > 0$ and $\mathrm{mod}(I, I_0)$ means the value of I modula I_0.

The image is on figure 4.

c) Sinusoidal probability functions:

Here we take:

$$p_k^R(I,J) = \frac{(a_k \mid \sin I \mid + b_k \mid \cos J \mid + \alpha_k)}{\sum_{i=1}^{4} p_i^R(I,J)}$$

$a_k, b_k \geq 0, \alpha_k > 0$. We obtain the texture of figure 5.

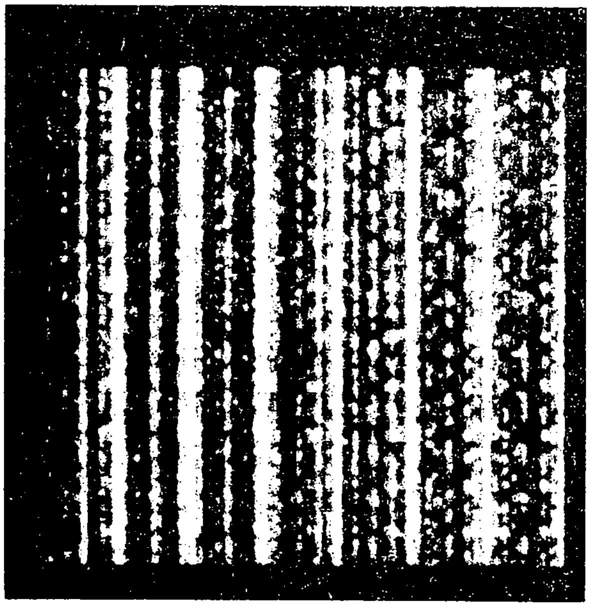

Figure 4: texture obtained with congruent probabilities

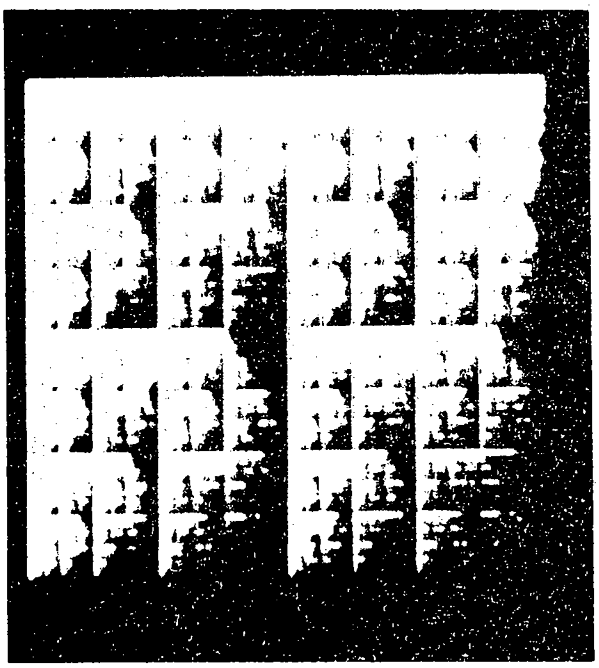

Figure 5: texture obtained with sine polynomials probabilities

d) "Special Effects":

- Border favorising texture:

We take:

$$p_k^R(I,J) = \frac{\sqrt{i_k^2 + j_k^2}}{\sum_{i=1}^4 p_i^R(I,J)}$$

where $(i_k, j_k) = w_k(I,J)$.

$p_k^R(I,J)$ is proportional to the modulus of the image of (I,J). Statistically, at each step of the process, we shall choose the function that sends the point the furthest away from the center, and then we shall favorise the borders.

The result is on figure 6.

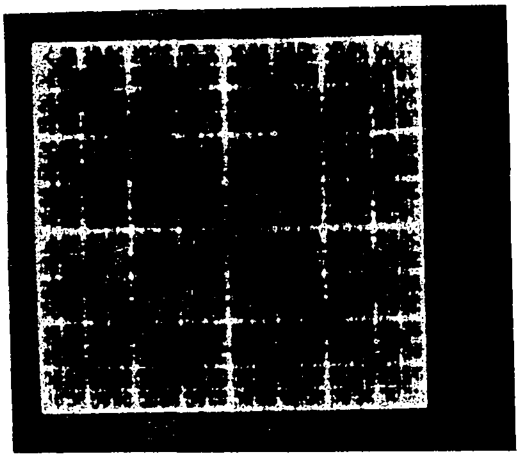

Figure 6: border favorising texture

- Center favorising texture:

Conversely, if we take:

$$P_k^R(I,J) = \frac{(\sqrt{i_k^2 + j_k^2})^{-1}}{\sum_{i=1}^4 p_k^R(I,J)}$$

we shall favorise the center, and obtain figure 7.

- "Turning" texture:

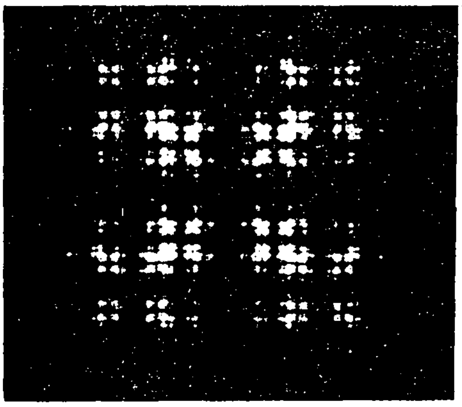

Figure 7: center favorising texture

If we want to generate a "turning texture" around the square we shall take:

$$\text{if} \quad I > 0, J > 0 \quad p_k^R(I,J) = a_k$$

$$\text{if} \quad I > 0, J < 0 \quad p_k^R(I,J) = b_k$$

$$\text{if} \quad I < 0, J < 0 \quad p_k^R(I,J) = c_k$$

$$\text{if} \quad I < 0, J > 0 \quad p_k^R(I,J) = d_k$$

with:

$$\sum_{i=1}^{4} a_k = \sum_{i=1}^{4} b_k = \sum_{i=1}^{4} c_k = \sum_{i=1}^{4} d_k = 1$$

and each $a_k, b_k, c_k, d_k > 0$.

Finally, on each line i of the matrix M:

$$M = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \\ d_1 & d_2 & d_3 & d_4 \end{pmatrix}$$

the $(i+1)$ term modula 4 is significantly greater than the 3 others.

Resulting textures are on figure 8, 9, 10.

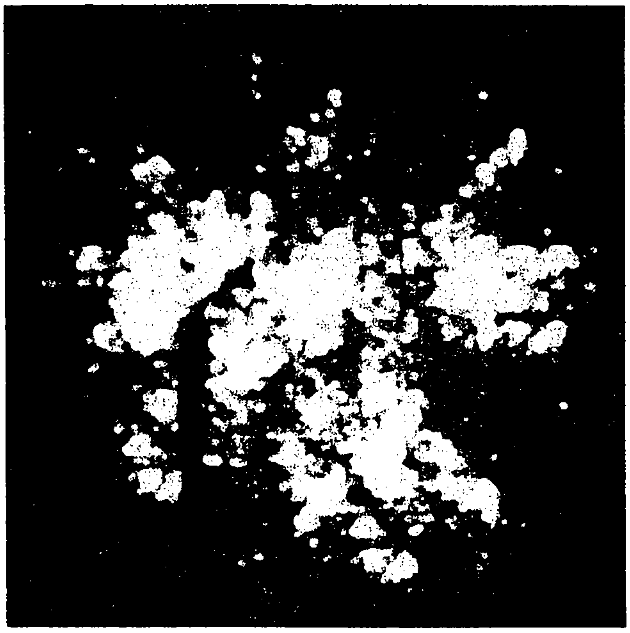

Figure 8: turning texture 1

4.2 Tests on Natural Textures

To validate our approach, we need now to try our formula on natural images. Three types of textures have been tried :

- aerial photographs of the ground.
- Brodatz texture of wood.
- mammography.

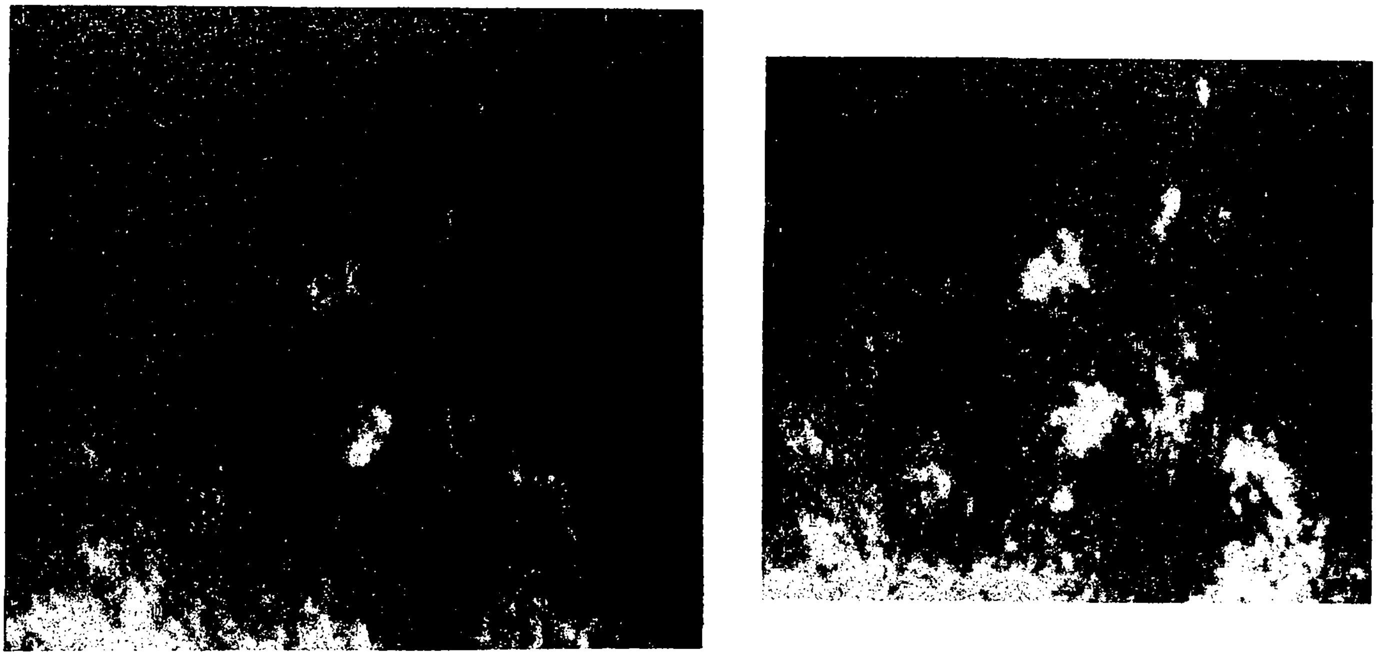

Figure 13: mammography and reconstructed image

However, other kinds of images, such like the ones on 14 and 15 are impossible to generate with our algorithm

To understand why, we must have a deeper understanding of the method.

4.3 Analysis of the texture coding scheme

Two problems arise :

(P1) : For a given texture, is it always possible to analyse it with the fractal probabilities method, and if not, what is the criteria of possibility ?

(P2) : Suppose that T is an analysable texture, what we get are four probability images of size half the size of the initial texture. Hence, we have not gain anything in terms of data compression or even in terms of image analysis, since we have exactly the same amount of data to process.

In fact, problems (P1) and (P2) are linked : if texture T "resembles" a fractal texture, the probability images given by formula 6 will indeed encode T and thus will allow to synthetise it. Furthermore, these probability images will be much simpler than the original texture, hence will

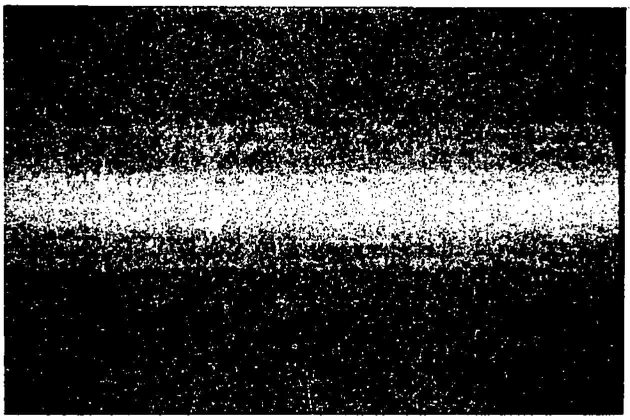

Figure 14: an impossible image to reconstruct

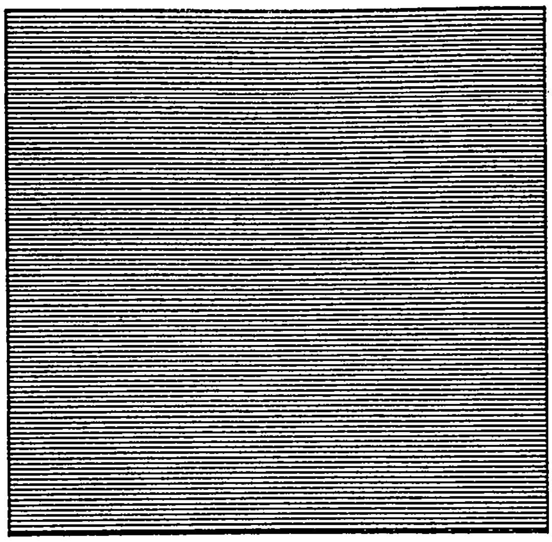

Figure 15: a second impossible image to reconstruct

be easier to compress and analyze.

4.3.1 Problem (P1)

What are the possible reasons for formula 6 not to hold, that is not to give a correct encoding of the texture ?

Theoritically, the only restriction we have made concerns overlapping : formula 6 ceases to be correct when a pixel (I, J) have more than one inverse image (i, j).

If the texture has been generated with some w_i such that there exists a region B (with strictly positive measure) such that :

$$w_i(A) \cap w_j(A) = B$$

for the attractor A, then we do not know how to compute p_i and p_j on this region, but only the sum $p_i + p_j$. This happens to be the case on figure 14.

This theoritical condition being stated, how can we tell visually that our texture T fits within this framework ?

To answer that , we have to remember that formula 6 is based on a change of resolution of the image. The assumption that T is fractal is loosely equivalent to the following assumption :

The image T_N of T at resolution N and the image T_{2N} of T at resolution $2N$ are not dramatically different.

Of course, this rule is somewhat vague, but it helps to tell very quickly if our theory will apply to the considered texture. For instance, the image shown on figure 13 changes dramatically if we divide the resolution by 2 : the thin black and white lines will disappear leaving the place for an uniform grey image.

A more precise way to forecast the results of the fractal probability encoding is to measure wether the texture is fractal or not. We can do this by simply using classical methods for estimating the fractal dimension (box counting method, morphological method, variation method, Voss' method, etc...), and see if the points obtained on the graph $Log(N)$ versus $Log(1/\varepsilon)$ lie on a "reasonably" straight line. If so, the method is valid, since the texture will possess a scale invariance property.

4.3.2 Problem (P2)

If the image is nearly fractal, then we shall obtain four probability images that encode well our original texture T.

Moreover, since T is fractal, our probabilistic analysis will be well adapted, and generally the probability images are much more simpler, which means easier to compress and understand that the original image : this is because we have decomposed T in a way that reflect its construction.

To verify this assumption, we consider the problem of coding the probability functions of the textures displayed on figure 3 to 13.

Concerning the probability images of all synthetic textures (figure 3 to 10), we have been able to reconstruct exactly the formula for all $p_i's$ given the types of the functions used (polynomials, sinusoidals, etc...) with a simple gradient method.

For the natural textures of figure 11, 12 and 13, we have tried and approximate the probability images by polynomial functions of two variables, using a least square minimisation method.

For figures 11 and 12, we obtained good fittings using 12th degree functions in X and Y, that needs 676 coefficients for the encoding.

We might remark here that classical methods for texture coding give better results on those types of textures, in terms of the number of parameters needed for reconstruction. For instance, autocovariance methods for the analysis-synthesis of such a texture give good results with only 40 coefficients (see [8]).

But it is important to notice that the two algorithms (autocovariance and IFS) do NOT do the same thing at all:

While the autocovariance method allows to encode and synthetise a texture that only looks like the original one, the fractal probability functions method reconstruct the exact image T except for a small number of noisy points (which should theoritically have zero measure).

Exact data compression methods, like run length encoding, of course give very poor results on fractal images, since the grey levels changes very erratically here.

For texture 13, we obtained a good fitting using 8th degree functions in X and Y, which is even better than on the previous texture. On image 12, autocovariance techniques do not work well, because it is not homogeneous enough.

This image then shows that most classical methods, like for instance autocovariance, would not be well adapted to perform an analysis on those type of textures. On the contrary, fractal probability functions method allows such an analysis: when presented with different mammographic images, analysis using the compressed fractal probability functions will be reliable (because the encoding is well done) and easy (since the data are very much compressed).

A more complete study on this very subject remains to be done, to show the possibilities and limits of the method.

5 Conclusion

In this work, we have shown how to encode fractal textures with probability functions given by the IFS theory. This encoding allows data compression, since it is possible to reconstruct almost exactly the original image from the probabilities. It also open the way for new methods for texture segmentation. Further work in this area is currently done to explore the limits of this approach.

References

[1] M. Barnsley. Fractal functions and interpolation. *Constructive Approximation*, 1985.

[2] M. Barnsley and S. Demko. Iterated function system and the global construction of fractals. *Proceedings of the Royal Society*, A 399:243-245, 1985.

[3] M. Barnsley, S. Demko, J. Elton, and Geronimo J. Invariant measures for markov processes arising from iterated function systems with place-dependent probabilities. Georgia Tech. preprint.

[4] M. Barnsley, V. Ervin, D. Hardin, and J. Lancaster. Solution of an inverse problem for fractals and other sets. *Proc. Natl. Acad. Sci. USA*, 83, 1986.

[5] S. Demko, L. Hodges, and B. Naylor. Construction of fractal objects with ifs. *Siggraph*, 19(D 3), 1985.

[6] W. Doeblin and R. Fortet. Sur des chaînes à liaisons complètes. *Bull. Soc. Math. de France*, 65:132-148, 1937.

[7] J.H. Elton. An ergodic theorem for iterated maps. In *Georgia Tech. preprint*, 1986.

[8] André Gagalowicz and Christine Graffigne. Blind texture segmentation. In *Proc. International Conference on Pattern Recognition*. IEEE, October 1988. Pékin, Chine.

[9] J. Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30:713-747, 1981.

[10] J. Levy Vehel. About lacunarity, some links between fractal and integral geometry, and an application to texture segmentation. In *ICCV*, 1990.

[11] J. Levy Vehel and A. Gagalowicz. Shape approximation by a fractal model. In *EUROGRAPHICS*, 1987.

[12] J. Levy Vehel and A. Gagalowicz. Fractal approximation of 2d objects. In *EUROGRAPHICS*, 1988.

[13] B.B Mandelbrot. *Fractals: Form, Chance and Dimension.* CA: Freeman, San Fransisco, 1977.

[14] B.B Mandelbrot. *The Fractal Geometry of Nature.* CA: Freeman, San Fransisco, 1982.

[15] H.O. Peitgen and D. Saupe. The science of fractal images. In *Springer Verlag*, 1988. New York.

[16] S. Peleg, J. Naor, R. Hartley, and D. Avnir. Multiple resolution texture analysis and classification. *IEEE*, PAMI-6(4), July 1984.

[17] R.F. Voss. Random fractal forgeries. In R.A. Earnshaw, editor, *Computer Graphics.* Springer-verlag Berlin Heidelberg, 1985.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique.

ISSN 0249-6399