# Texture and multifractals: new tools for image analysis

Jacques Lévy Véhel, Pascal Mignot, Jean-Paul Berroir

UNITÉ 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
25ème anniversaire
N° 1706

*Programme 4*
*Robotique, Image et Vision*

# TEXTURE AND MULTIFRACTALS : NEW TOOLS FOR IMAGE ANALYSIS

Jacques LEVY VEHEL
Pascal MIGNOT
Jean-Paul BERROIR

Juin 1992

# Texture and Multifractals: New Tools for Image Analysis

# Texture et Multifractals: de nouveaux outils pour l'Analyse d'Image

Thème 4

**Jacques LEVY VEHEL, Pascal MIGNOT, Jean-Paul BERROIR**
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**

We present some new ideas for image analysis and texture segmentation. The two important points are the introduction of the multifractal theory for image understanding in a rigorous manner, which allows to do all the computations directly on the discrete signal, and the definition of a precise framework for texture analysis, in which we do not choose any a priori model for the study but instead let the peculiar data in each case decide which model is the most appropriate. We define an approach general enough as to handle a great variety of textures and we propose a new scheme, based on learning and optimal cooperation, that gives good results on a lot of segmentation experiments presented here.

Nous présentons de nouvelles idées pour l'analyse d'images et la segmentation de textures. Les deux points importants sont l'introduction rigoureuse de la théorie multifractale pour la compréhension des images, qui permet de faire tous les calculs directement sur le signal discret, et la définition d'un cadre précis pour l'analyse de textures, dans lequel aucun modèle à priori n'a besoin d'être spécifié. Nous proposons une approche suffisamment générale pour travailler sur une grande variété de textures, et nous présentons un nouveau schéma, basé sur une approche par apprentissage et coopération optimale, qui donne de bons résultats sur plusieurs tests de segmentation présentés ici.

# 1 Introduction

In this work, we propose some new ideas for image analysis using texture and multifractal paradigms. We first present the multifractal theory and its applications to image description, and we show that this approach allows to work directly on the discrete signal. We then introduce a system for texture classification, ARTHUR, which is based on a learning scheme and does not make use of any a priori model. We finally move to image segmentation by means of an extension of ARTHUR that uses the notion of mixed classes to allow accurate textures boundaries detection on complex images.

# 2 Multifractal Approach

Though fractal geometry has been introduced a long time ago in image analysis, it is not yet used extensively ([Man82], [B.M77], [B.D88], [BD85], [LV91], [BEHL86], [Hut81], [PS88]).
Some authors have used fractal dimension to perform texture classification and image segmentation ([PNHA84],[Pen84]), other have used higher order dimensions or measures, as correlation or lacunarity ([KCC89],[LV90]), to refine the results and have obtained some interesting results. Very few papers have been devoted to the use of multifractals in image analysis ([AKA88]), but multifractals are intensively studied in physics, meteorology and other fields ([LDG+90], [Oon89], [Kah90]).

The main point that justifies the introduction of multifractals in image analysis is the following one :
Fractal dimension is a nice tool for characterizing the irregularity of a curve or a surface. Though its measurement is not very precise on images, it is fast to compute and can sometimes help to get specific features of the data. However, it seems to us that applying it to characterize an image is totally unfounded. This approach assumes that the 2D grey level image can be seen as a 3D surface, or, equivalently, that the grey levels can be assimilated to a spatial coordinate on the z-axis. This assumption has no theoretical basis and we believe that it leads to a fundamentally false analysis of the image. Instead, we should look at the grey levels as a measure, laid upon a generally compact set, totally unhomegeneous to space coordinates.[1]
In this framework, we may now appeal to fractal tools to solve some specific problems. The correspondent of fractal dimension (which is a notion that refers to set) for measures are the multifractal measures. They have been developed by Hentschel, Proccacia and Mandelbrot ([I.P], [HI83], [Shr86], [Man], [Man89]).
The first approach to multifractals is the following one :
Let $f$ be a summable function from $[a, b]$ to . Let $\ell$ be an natural number and assume that we divide $[a, b]$ into $\ell$ intervals of equal length $1/\ell$. Let $\mu$ be any measure defined from $f$ on each interval $k_i(\ell)$. For instance, we may take:

$$\mu(k_i) = \int_{k_i(\ell)} f(t)dt$$

The measure $\mu$ will be called multifractal iff :
a)

$$\forall k_i(\ell), \exists \alpha_{k_i} \in \ ^{+} / \ \mu(k_i(\ell)) \sim \left(\frac{1}{\ell}\right)^{\alpha_{k_i}}$$

when $\ell^{-1} \longrightarrow 0$
which means that $\mu(k_i(\ell))$ is equivalent to $(\frac{1}{\ell})^{\alpha_{k_i}}$
$\alpha$ is called the Hölder exponent.
b)
Let $E_{\alpha_0} = \{k_i(\ell)/\alpha_0 - d\alpha < \alpha_{k_i} < \alpha_0 + d\alpha\}$
for a certain (small) $d\alpha$ and let

$$N_\alpha = \text{ card } (E_\alpha).$$

[1]Pentland (see [Pen84]) has shown that, under certain conditions, the image of a 3D fractal surface is also a fractal, but this is not relevant here, since we are talking of a different problem

Then :

$$N_\alpha \sim d\alpha(\frac{1}{\ell})^{-g(\alpha)}$$

where $g$ is a function of $\alpha$.

Intuitively, this means that when we group all "points" with same $\alpha$, we get a set $E_\alpha$, whose fractal dimension is given by $g(\alpha)$.

The $(\alpha, g(\alpha))$ description is then obtained, loosely speaking, in splitting the interval $[a, b]$ into subsets where all the points have same Hölder exponent (some authors speak about strength of singularity for the measure) and then characterizing each subset by its fractal dimension. This approach includes both local (via $\alpha$) and global (via g) informations.

In the second approach, we directly compute some sorts of "moments" or generalized dimensions of our measure. The generalized dimension of order $q, q \in$ , is defined by :

$$D_q = \frac{1}{q-1} \lim_{\ell \to \infty} \frac{\log \sum_{i=1}^{\ell} \mu^q(k_i)}{\log(\ell^{-1})}$$

with the same notations as above.

People sometimes use $\tau(q) = (q-1)D_q$ instead of $D_q$.

It may be noticed that $D_0$ corresponds to the fractal dimension of the support of the measure, $D_1$ (which is defined by continuity when $q \to 1$) to the information dimension, and $D_2$ to the correlation dimension. The description given by $(q, D_q)$ is global since we sum over the whole interval.

Intuitively, when $q$ is very high, $D_q$ is sensitive to those parts of the interval where the measure is very dense, as when $q$ is very low (meaning negative with high absolute value), we get information on the sparse regions with respect to the measure.

The two descriptions $(\alpha, g(\alpha))$ and $(q, D_q)$ are dual and they have some profound links with thermodynamics, of which we shall only mention the Legendre transform. $\alpha$ might be compared to $U$ (energy) $q$ to $1/T$ (temperature) $g$ to $S$ (entropy) and $\tau$ to $F$ (free energy).

They are couples of conjugate variables and we have :

$$\begin{cases} \alpha = \dfrac{d\tau}{dq} \\ g = q\alpha - \tau(q) \end{cases} \Longleftrightarrow \begin{cases} q = \dfrac{dg}{d\alpha} \\ \tau = q\alpha - g(\alpha) \end{cases}$$

(Legendre transform )

For image analysis, we shall add to the classic theory the measurement of $\ell(\alpha)$ which is the lacunarity of the subset $E_\alpha$. Generally speaking, lacunarity measures the homogeneity of a fractal distribution. It is defined as the variance of the ratio $\frac{m'}{m}$, where $m$ is the mean mass (intensity) in a fixed size window and $m'$ the actual mass in the window when it is moved over the whole set ([LV91])

$$\ell = < (\frac{m'}{m} - 1)^2 >$$

we shall make a description of images in terms of $(\alpha, g(\alpha), \ell(\alpha))$.

To have a little more insight of what $\alpha$ might mean, we give some examples. We shall use the frame of Non Standard Analysis (NSA), since it simplifies very much the notations. However we shall not go into any profound development in this field, so that the reader who does not know about Non Standard Analysis will not have any effort to make. For our purposes, it suffices to know that in NSA, a set $^*$ is defined, which is a non-archimedean extension of ( i.e there exists numbers $x$ and $\omega$ such that $\forall n \in$ , $nx < \omega$) and which contains both infinitely large and infinitely small numbers. Let $\omega$ be such a infinitely large number.

Then: $\forall x \in$ , $\forall n \in$ $\quad nx < \omega$

We shall say that $x \ll \omega$, or that $\frac{x}{\omega}$ is infinitely close to zero. Most of the rules of can be applied, with certain care, to $^*$ .

Now let us begin with the $1D$ case. Let $f$ be a function from $^*$ to $^*$ and let $[a, b]$ be an interval of .

We note $x_0 = a$,

$$x_i = a + \frac{i}{\omega}(b-a), \ \ i = 1, \omega - 1$$

$$x_\omega = b$$

It can then be shown under certain conditions that the quantities :

$$\sum_{i=1}^{\omega} f(x_i)\frac{b-a}{\omega}$$

and

$$\int_a^b f(x)dx$$

are infinitely close (their difference in infinitely small)

We shall write

$$\int_a^b f(x)dx \simeq \sum_{i=1}^{\omega} f(x_i)\frac{b-a}{\omega} = \frac{b-a}{\omega}\sum_{i=1}^{\omega} f(x_i)$$

If interval [a,b] has Hölder exponent $\alpha$ with respect to $f$, we may write, with $t = b - a$ :

$$k(b-a)^\alpha \simeq \frac{b-a}{\omega}\sum_{i=1}^{\omega} f(x_i), k \in$$

$$kt^{\alpha-1} = \frac{\sum_{i=1}^{\omega} f(x_i)}{\omega}$$

$$\alpha \simeq \frac{\log\sum_{i=1}^{\omega} f(x_i) - \log\omega - \log k}{\log t} + 1$$

Assume $f$ is constant : $f(x) = c, \forall x \in [a, b]$

$$\alpha = 1 + \frac{\log c\omega - \log\omega - \log k}{\log t}$$

$$\alpha = 1 + \frac{\log c - \log k}{\log t}$$

Since $t \simeq 0$ and $t \neq 0$ : $\alpha \simeq 1$

If $f(x) = x, \forall x \in [a, b]$ we find

$$\log\omega a < \log\sum_{i=1}^{\omega} x_i < \log\omega b$$

Thus

$$\frac{\log a - \log k}{\log t} < \alpha - 1 < \frac{\log b - \log k}{\log t}$$

and again, since $t \simeq 0, t \neq 0$, we have : $\alpha \simeq 1$.

It is easy to see that this result holds for any polynomial function $f$

Now if $f$ is a two steps function :

$$f(x) = c \text{ if } x \in [a, u]$$
$$f(x) = d \text{ if } x \in [u, b]$$

we get

$$\alpha \simeq 1 + \frac{\log\left(\sum_{i=1}^{\omega_u} x_i + \sum_{i=\omega_{u+1}}^{\omega}\right) - \log\omega - \log k}{\log t}$$

with

$$u = a + t\frac{\omega_u}{\omega}$$

and then $\frac{\omega_u}{\omega} = \frac{x-a}{b-a} = p$ which is a standard number (an element of ).

Thus

$$\dot{\alpha} \simeq 1 + \frac{\log[pc + (1-p)d] - \log k}{\log t}$$

and again $\alpha \simeq 1$

This shows that one discontinuity and even a zero measure set of discontinuities does not affect the value of $\alpha$. This is very natural since the multifractal approach is indeed designed to handle the case of very irregular measures, strange to Lebesgue measure.

In 2D the computation is very simple too

$$\int\int f(x,y)dxdy \simeq \sum_{i=1}^{\omega}\sum_{j=1}^{\omega} f(x_i, y_j)\frac{t^2}{\omega^2}$$

with the same notations as above.

By definition of $\alpha$

$$kt^{\alpha} \simeq \frac{t^2}{\omega^2}\sum_{i=1}^{\omega}\sum_{j=1}^{\omega} f(x_i, y_j)$$

and thus

$$\alpha \simeq 2 + \frac{\log\sum_{i=1}^{\omega}\sum_{j=1}^{\omega} f(x_i, y_j) - 2\log\omega - \log k}{\log t}$$

Again for polynomial $f$ we find that $\alpha \simeq 2$.

Before we move to a truly multifractal measure, although very simple, let us just mention that the function $f$ :
$f(x) = ax^{a-1}$, with $x \in \ ^{+\star}$ and $a \in ]0,1[$, has $\alpha = 1$ everywhere except when $x \simeq 0, x \neq 0$, where $\alpha = a$
Now let us to move to binomial measures. To construct a binomial measure on $[0, 1]$, we first divide it into $[0, \frac{1}{2}]$ and $]\frac{1}{2}, 1]$ and we set "mass" $m_1$ to $[0, \frac{1}{2}]$ and $m_2$ to $]\frac{1}{2}, 1]$, with $m_1 + m_2 = 1$. We then iterate the process in both subintervals, getting $m_1^2$ in $[0, \frac{1}{4}], m_1 m_2$ in $]\frac{1}{4}, \frac{1}{2}]$, etc ...

Finally, if we have divided $[0,1]$ into $2^\omega$ intervals $[t_i, t_{1+1}]$ with :

$$t_{i+1} = t_i + \frac{1}{2^\omega} = \frac{i}{2^\omega}$$

we may write

$$\forall x_i \in [t_i, t_{i+1}[, x_i = 0.t_i^0 t_i^1 \dots t_i^{2\omega} + dx_i$$

where $dx_i \in [0, \frac{1}{2^\omega}[$, and $t_i^k = 0$ or 1. Upon $[t_i, t_{i+1}[$ lies the mass $m_0^{\varphi_i^0} m_1^{\varphi_i^0}$, if $\varphi_i^0$ is the number of times we have a 0 in the decomposition of $x_i$ and $\varphi_i^1$ the number of 1. It is then easy to compute $\alpha, g(\alpha)$ and $\ell(\alpha)$ which are given by :

$$\alpha_i = -\varphi_0 \log_2 m_0 - \varphi_1 \log_2 m_1$$

$$g(\alpha_i) = -\varphi_0 \log_2 \varphi_0 - \varphi_1 \log_2 \varphi_1$$

$$\ell(\alpha_i) = 2^{-\omega}\left[C_\omega^{\omega\varphi_0}\left(\frac{m_0^{\omega\varphi_0} m_1^{\omega\varphi_1}}{2^{-\omega}} - 1\right) + C_\omega^{\omega\varphi_1}\right]$$

The computation of $D_q$ gives :

$$D_q = \frac{1}{q-1}\log_2[m_0^q + m_1^q]$$

Examples of $g(\alpha)$ and $D_q$ are shown on figures 1. An $l(\alpha)$ curve is shown on figure 2.

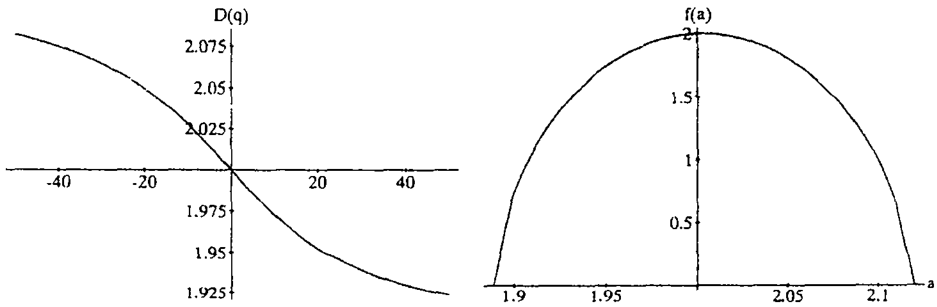


Figure 1: left : $D_q$ right : $g(\alpha)$ spectrum

The binomial case, besides the advantage that the multifractal functions can be exactly computed, teaches us the following fact :
If we need a continuous description of our signal, we may consider a step function taking constant values on each sub-interval defined by the discrete data, and equal to these data. For instance, if we denote by $f$ the step function that associates to each $x \in [t_i, t_{i+1}[$ the mass laid upon $[t_i, t_{i+1}[$, we get :

$$y = f(x) = m_0^{\omega\varphi_0} m_1^{\omega\varphi_1}$$

and :

$$\int_{t_i}^{t+1} f(x)dx = \frac{m_0^{\omega\varphi_0} m_1^{\omega\varphi_1}}{2^\omega} = y(t_{i+1}) - y(t_i) = dy$$

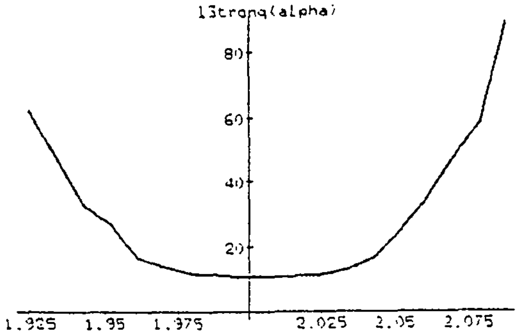


Figure 2: $l(\alpha)$ curve for a binomial measure

Thus : $dy = dx \times dx^{\alpha_i}$ with $dx = 2^{-\omega}$
finally : $dy = dx^{\alpha_i+1}$

The binomial case is the simplest case of a more general family, the multinomial measures. To construct such a measure on a given interval $I$ of euclidean dimension $D$, typically $[0,1]$ if $D=1$, $[0,1] \times [0,1]$ if $D=2$, we first divide $I$ into $N = b^D$ equal parts. For instance, a binomial measure in 2D will be generated by iterating the mass division on each fourth of $[0,1] \times [0,1]$, a quadrinomial measure in 1D by applying the same process on each fourth of $[0,1]$. Given $N$ multipliers $m_i, i = 1, N$ with $\sum_{i=1}^{N} m_i = 1$, we set mass $m_i$ on the $i^{th}$ subinterval, and then iterate the process on each subinterval. As before, $\alpha$ and $g(\alpha)$ can be easily computed by :

$$\alpha(\phi_i) = -\sum_{i=1}^{N} \phi_i \log_b m_i$$

$$g(\alpha(\phi_i)) = -\sum_{i=1}^{N} \phi_i \log_b \phi_i$$

If $N = 2$, as for the 1D binomial case, we have a parametric definition of the spectrum $(\alpha, g(\alpha))$. If $N > 2$ those two equations do not lead to a parametrization, for we only have two equations and $N$ variables. This means there exist several sets $E_{\phi_i}$ defined by a $(\phi_i)$ vector leading to the same singularity $\alpha(\phi_i)$ but differing with each other by their fractal dimension. $g(\alpha)$ is the fractal dimension of $\bigcup_{i=1}^{N} E_{\phi_i}, \alpha(\phi_i) = \alpha$. The fractal dimension of union of sets if the greatest dimension of each set, thus :

$$-G(\phi_i) = g(\alpha) = \sup_{\phi_i}(-\sum_{i=1}^{N} \phi_i \log_b \phi_i)$$

To compute $g(\alpha)$, we then have to solve $\min_{\phi_i} G$, given two constraints :

$$C_1(\phi_i) = \sum_{i=1}^{N} \phi_i - 1 = 0$$

$$C_2(\phi_i) = -\alpha(\phi_i) + \sum_{i=1}^{N} \phi_i \log_b m_i = 0$$

We then introduce the Lagrange multiplyers $\lambda_1, \lambda_2$ and we have to solve :

$$\nabla G + \lambda_1 \nabla C_1 + \lambda_2 \nabla C_2 = 0$$

which leads to a $(N+2)$ equations system with $(N+2)$ unknowns $\phi_i, \lambda_1, \lambda_2$ :

$$\begin{cases} \forall i, \log_b \phi_i + \lambda_1 + \lambda_2 \log_b m_i + \frac{1}{\log b} = 0 \\ \sum \phi_i - 1 = 0 \\ \sum \phi_i \log_b m_i + \alpha = 0 \end{cases}$$

Each unknown $\phi_i$ can be written as a function of $\lambda_2$ :

$$\phi_i = \frac{m_i^{-\lambda_2}}{\sum_{j=1}^{N} m_j^{-\lambda_2}}$$

Substituing $\phi_i$ in the last equation of the system leads to an equation with only one unknown, $\lambda_2$ :

$$\Lambda(\lambda_2) = \frac{\sum_{j=1}^{N} m_j^{-\lambda_2} \log_b m_i}{\sum_{j=1}^{N} m_j^{-\lambda_2}} = -\alpha$$

$\Lambda$ being a decreasing function of $\lambda_2$, we can find the solution iteratively. Nevertheless, two reasons prevent us to handle the problem that way : as $\lim_{\lambda_2 \to -\infty} \Lambda(\lambda_2) = -\alpha_{min}$ and $\lim_{\lambda_2 \to \infty} \Lambda(\lambda_2) = -\alpha_{max}$, values of $\alpha$ close to its extremum leads to a poor precision on $\lambda_2$, and secondly, $\Lambda$ is highly dependant on $(m_i)$, that is, a little variation of $(m_i)$ leads to a great variation of $\Lambda$, introducing important errors on $\lambda_2$. A better conditioning of this problem can be achieved by introducing a new function : if $m$ and $M$ are $\inf m_i$ and $\sup m_i$, let us define :

$$\Lambda^*(\lambda_2) = \frac{m^{-\lambda_2} \log_b m + M^{-\lambda_2} \log_b M}{m^{-\lambda_2} + M^{-\lambda_2}}$$

The reciprocal of $\Lambda^*$ is :

$$\Lambda^{*-1}(a) = \frac{\log - \frac{a + \log_b M}{a + \log_b m}}{\log \frac{M}{m}}$$

$\Lambda^*$ presents several advantages :

- It has the same variation domain as $\Lambda$.
- It has a reciprocal.
- It has the same sensitiveness to $m_i$ as $\Lambda$, thus the function $\Lambda(\Lambda^{*-1})$ is rather insensitive to $m_i$, as illustrated by figure 3, which represents graphs of $\Lambda$ and $\Lambda^*$ for two 2D-binomial measures with weights $(0.5, 0.2, 0.19, 0.11)$ for the dashed curve and $(0.27, 0.26, 0.24, 0.23)$ for the other one, and figure 4 which represents graphs of $\Lambda(\Lambda^{*-1})$ for the same measures.

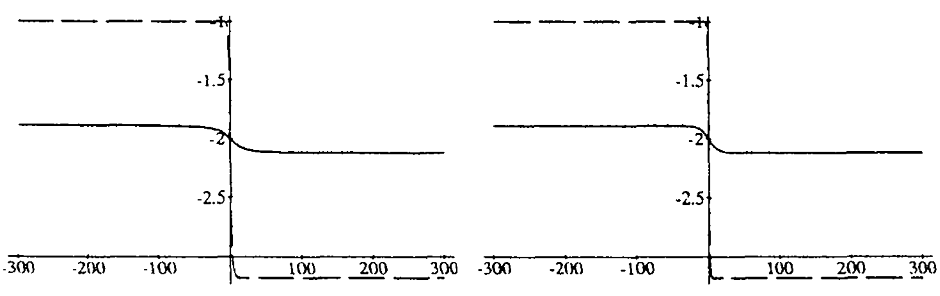


Figure 3: left : $\Lambda$ right : $\Lambda^*$

Computing the singularity spectrum $g(\alpha)$ of a $b$-nomial measure in any dimension then consists of those steps, for any value of $\alpha \in \ \alpha_{min}, \alpha_{max}\ $, with $\alpha_{min} = -\log_b M$ and $\alpha_{max} = -\log_b m$ :

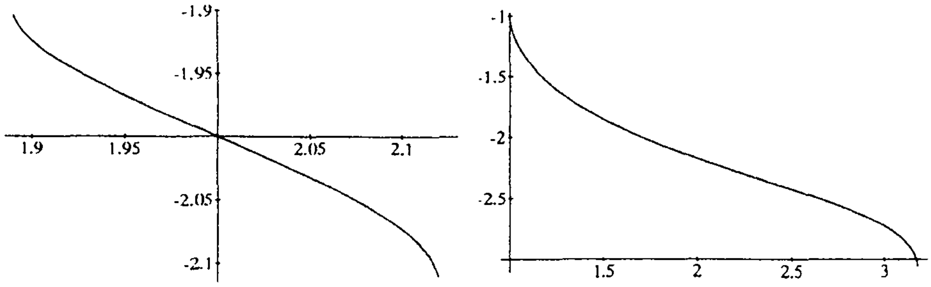


Figure 4: $\Lambda(\Lambda^{*-1})$

- Given $\alpha$, solve $\Lambda(\Lambda^{*-1}(a)) = -\alpha$.
- Compute $\lambda_2$ through $\lambda_2 = \Lambda^{*-1}(a)$.
- Compute each $\phi_i$ through $\phi_i = \frac{m_i^{-\lambda_2}}{\sum_{j=1}^{N} m_j^{-\lambda_2}}$.
- At last, $g(\alpha) = -\sum_{i=1}^{N} \phi_i \log_b \phi_i$.

The interest of multinomial measures for image analysis is that it is a large family of multifractal distributions whose spectrum can be computed exactly. Since some methods are available for computing with good precision the spectrum of any image by use of a specific related multinomial one (see [LVB91]), we are able to get reliable information on the image through its $(\alpha, g(\alpha), l(\alpha))$ description. Indeed, direct computations generally give poor results, unusable for any interpretation. Instead, using multinomial measures allows us to bring the power of multifractal analysis into image understanding : for instance, image points having $\alpha \simeq 2$ will be points where the measure is regular, thus where no large changes appear. Points with $\alpha << 2$ or $\alpha >> 2$ will be included either in zones with high "gradient", or in zones of discontinuity of the signal or of its derivative. $\alpha \neq 2$ thus allows to mark points where "something is hapenning", as $\alpha = 2$ indicates smooth behaviour or very small activity. On figure 5, original image of Lena is shown along with 3 iso-$\alpha$ images. The first and the third ones ($\alpha = 1.8$ and $\alpha = 2.2$) contain active points, while the second one ($\alpha = 2$) exhibits points where the signal remains smooth.

Figure 5: Lena and $\alpha$ images of Lena

On figures 6,7, and 8 are displayed an original image (spot image, office, and MR image of the brain), the binary image of points whose $g(\alpha)$ is smaller than a certain threshold (which is equivalent to $\alpha << 2$ or $\alpha >> 2$) and, for comparison, the edges obtained with a classical Canny-Deriche detector. The $g(\alpha)$ images show how the multifractal process, though failing to detect certain edges, captures a more global description of the scene, giving for instance always closed "edges", that define regions. We believe that a more profound study of this approach could provide an interesting alternative for edge detection, in which one would not have to go through a whole process of smoothing the original discrete image signal before extracting the contours

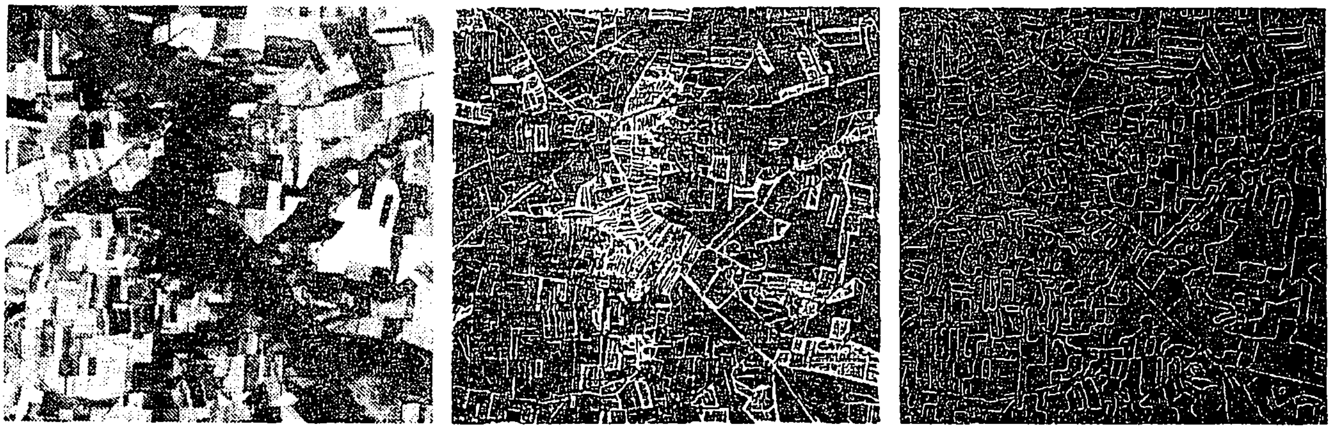

Figure 6: spot image, points with $g(\alpha)$ lower than 1.8, Canny's edges

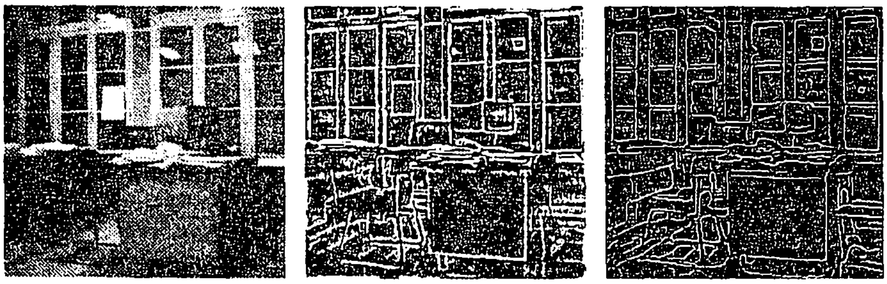

Figure 7: office image, points with $g(\alpha)$ lower than 1.8, Canny's edges

# 3 Texture analysis : basic ideas

Much work have been done on texture analysis in the last decades. The authors have addressed the problem of classification, compression, or segmentation using several approaches ([Har79], [WDR76], [VP87], [AR81], [LV88], [PNHA84], [Sul], [FK80], [BCG90], [Gal75], [KCC89], [GG88], [LV90], [MC91],[FJ91],[PIP91],[LV91]).

Several trends can be distinguished : supervised versus unsupervised methods, model-based versus "general" methods, methods using multiresolution approaches or not.

Unsupervised approaches, for instance, make no use of a priori knowledge of the texture types to analyze. Model-based methods often look at the texture as a realization of Markov Random

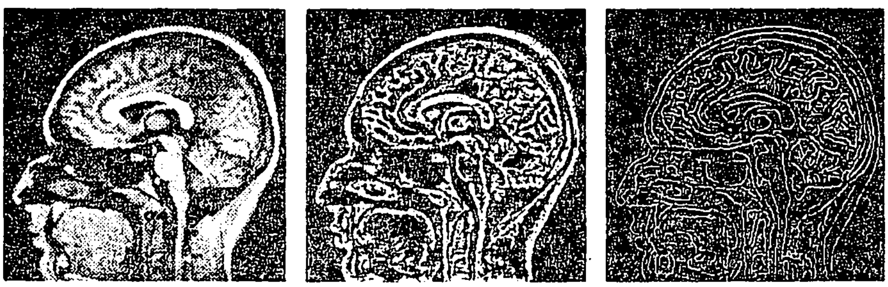

Figure 8: MR image of the brain, points with $g(\alpha)$ lower than 1.8, Canny's edges

Field, stationary or not, or of a Fractionnal Brownian Motion. Multiresolution approaches use descriptions of the image at several levels, most of the times in the frame of Wavelets theory.

Our method is a supervised one, without any model assumptions. The justifications for these choices are the following ones :

- one may believe that the human brain recognizes textures without learning : after all, one an classify textures never seen before on the basis of lower level informations such as coarseness, orientation, etc ..., and experiments have longly shown that specialized cells in the cortex react to precisely oriented line segments, motion, etc ...

  However, at least two reasons stand for a learning scheme :

  - If it is true that humans recognize textures with lower level mechanisms, those have been learnt at some time. Then, we should show the computer how to do that. Since we do have to do a learning, it is easier and more reliable to teach the computer higher level knowledge as the texture type itself.
  - It is a false impression that the human brain recognizes textures without learning : this is only true for a certain class, as outdoor natural textures, hand-made textures. This class is indeed very vast but does not include some "special" images, as for instance the cork textures displayed on figure 14, or medical images : echographies or simply radiographies may be simply impossible to interpret for a non trained person.

Figure 9: radiography of the breast with microcalcifications

  On figure 9 it is hard to detect what the radiologists call microcalcifications, which are small calcareous deposits in the breast. They look very much alike artefacts or simply high density zones of the image . This texture detection is a learnt ability. We don't believe that what radiologists learn to do on large training data sets, the computer can do without.

- Since we do training, we do not really need a model for our textures : in fact, there is no model which is general enough as to handle all possible textures : some are Markovian, some Fractals, some of another still unidentified type. Some may be well described with multiresolution techniques and others do not. Thus, it seems an impossible task to design a system able to recognize one texture among all possible textures in the universe. Instead, we shall restrict our work, in each case, to the recognition of textures among a finite number of patterns, which we shall call a library. Doing that, we shall let the system find by itself an "empirical " model to describe a peculiar set of textures. Notice that we may define and use as many libraries as we need.

- This leads us to the last point : a large number of features have been proposed for texture measurements. Some give good or excellent results, but only on a restricted class. There is no universal parameter. The idea is then to let the system compute a great variety among all of known features and allow it to choose a small subset that makes the best discrimination in the textures library.

Theses idcas will be used both for classification and segmentation.

# 4 Texture Classification

## 4.1 Basic ideas

The fundamental principles of ARTHUR, the texture classifier, are the following ones:

- **learning scheme** : we have explained our reasons for that approach in the previous section.
- **non hierarchical multi-parametric approach** : one of the main aims of ARTHUR is to be able to use jointly the discriminant powers of for instance, Haralick's energy and Fractal dimension. The system computes all the parameters and then let the data analysis module handle them.
- **robust data analysis** : several methods for data analysis have been included in ARTHUR, some classical, others developed specially for our problems. It is then possible, exactly as for the parameters, to choose the data analysis scheme that gives the best discrimination.
- **notion of library** : all the results given by ARTHUR (selected parameters, discrimination power, classification ) are local to a library.
  It has occurred in our experiments that for library { wood, grass, ivy }, parameter $p$ and $q$ are good, as for library {wood, clouds, water}, they behave poorly. Nevertheless, nothing prohibits to compare different libraries using the results of ARTHUR. Finally the system not only allows to classify textures in a library, but also to show the existence of large types of textures, as for instance Markovian or Fractal ones, according to the types of parameters that best describe them.
- **Qualitative description** : In addition to all quantitative results (values of the parameters for each sample, means, variances, Fisher test, correlations, decision scheme, classification, etc ...), we may have a qualitative insight consisting in :
  - a multiplot representation of the results.
  - a graphic display of the correlation between parameters
  - a decision tree

Finally we mention that ARTHUR is very easy to use, the user interface has been carefully studied. Besides this integration, our contribution here lies in the data analysis module and in the introduction of new parameters, namely Multifractal and Integral Geometry ones.

We shall describe the main parameters used in ARTHUR, then the data analysis module, and finally show some results.

## 4.2 The parameters

They have been separated into several classes :

### 4.2.1 Statistical parameters

- **Moments** : All sorts of moments have been implemented : centered and not centered, invariant, etc ..., up to the seventh order.
- **Autocovariance** up to a translation of six pixels length (see [GM85]).
- **Haralicks's parameters** : Here also, the translation ranges from 1 to 6 and all classical features are available (energy, entropy, contrast, maximum probability, correlation, etc ..., see [Har79])
- **Run lengths**: All classical run-lengths parameters have been implemented in the four directions (see [Gal75]).

### 4.2.2 Autoregressive model

A 3 parameters autoregressive model has been implemented

### 4.2.3 Parameters based on Fourier transform

All classical parameters based on the Fourier transform have been implemented (see [WDR76]).

### 4.2.4 Integral geometry

We make some brief recalls about Integral Geometry.

Let $G$ be a straight line in the plane. We can determine $G$ by the angle $\phi$ that it makes with a fixed direction ($0 \leq \phi < 2\pi$) and by its distance $p$ from a defined origin $O (p \geq 0)$. $(p, \phi)$ are the polar coordinates of the foot of the perpendicular from the origin onto the line.

We define :

$$dG = dp \wedge d\phi$$

This density is the only invariant measure for sets of lines (see [San76]).

The parameters we use in ARTHUR are defined by :

$$f(k) = \frac{1}{2} \int max(k, n(G)) dG$$

where $n(G)$ is the number of intersections between $G$ and the binary set $E$ that we are measuring, and the integration is taken over the plane. $f(k)$ is the Favard length of order $k$ of $E$. For generalization of $f$ to grey level sets and other results, see [LV90] and [LVB91].

$f(k)$ measures the "lengths" of slices of the texture taken at different levels defined by the intensity values and and also takes into account the default of convexity of these slices. They are robust parameters, fast to compute.

### 4.2.5 Multifractal parameters

We implemented the parameters described in section 2, namely the $(\alpha, g(\alpha), l(\alpha))$ and the $(q, D_q)$ descriptions of the image.

The user may finally choose between more than one thousand parameters.

## 4.3 Data Analysis

Once the library is defined (i.e. the classes are defined and some images are assigned to each class) and the parameters have been computed on every sample, the data analysis module takes on. Among the possible methods, we have chosen to focus on a peculiar one, that we have been designing specially for our type of problems. For description of other methods, see for instance [LVJ91],[CL82],[Ca90].

### 4.3.1 Introduction

We present here a new method for discrimination with the following features :

- the set of parameters is large
- the set of classes is large
- the a priori classes of the learning samples is known

The method proceeds in two steps :

- First step : Discrimination

  We reduce to several two classes problem. For each pair of classes, a discriminating space is chosen via a density estimator on the learning set and a criteria. We obtain a collection of discriminating spaces of small dimension.

- Second step : Decision.

  To classify an element $y$, we compute a vote matrix $M(y) = (D_{ij}(y))$, where $D_{ij}(y)$ is the probability for sample $y$ to be in class $C_i$ when opposed to class $C_j$. A decision function applied to $M(y)$ then leads to the final decision.

Notations :
Let $C = (C_i)$ be a set of $N$ classes, $E = (p_i)$ a set of $p$ parameters.
Let $L$ be the learning sample set, $x$ an element of $L$.
We associate to $x$ the couple $(V_x, C_x)$ where $V_x$ is a $^p$ vector (values of each parameter) and $C_x$ is the class of $x$
Let $S$ be a sample set to classify, $y$ an element of $S$
Let $(D_{ij})(y)$ be the vote function of the class $C_i$ versus the class $C_j$ for $y$ :

$$D_{ij} : S \longrightarrow [0, 1]$$

with $D_{ii}(y) = 1, \forall y$.
Let $F_d(y)$ be a decision function :

$$F_d(y) : M_{(N,N)}(y) \longrightarrow (C_i, i = 1,..N) \cup Ind$$

where $M_{(N,N)}(y)$ is a $N \times N$ matrix of vote functions and $Ind$ is the indecision vote. The i-th line of $M_{(N,N)}(y)$ collects all the votes on behalf of class $C_i$.

### 4.3.2 First Step : pair discrimination.

We have designed several ways to construct the discriminating spaces. We first present a general model, then a charaterization of this model in the gaussian case. However, any other model could be applied. The method allows to work locally on higher dimension spaces to improve the discrimination of a pair of classes when necessary.

1. General model : blind discrimination.

   The discrimination is blind in the sense that no model is assumed here. The method is based upon density estimation. We present the unidimensional model, but the method can be extended to higher dimension (in practice, choosing a dimension higher than 3 implies the use of a parametric model) with n-uplet of parameters. We assume that the costs of misclassification are the same for each class..

   a/ Density estimation.

   The density estimation is made on the learning set. We note $\tilde{\rho}_i^k$ the estimation of density function of class $C_i$ for parameter $k$. We implemented two kinds of density estimators, the simple histogram density estimator, and the kernel method estimator, which is more delicate to proceed (choice of h) but provides a higher estimation quality. It is based upon the following equation :

$$\tilde{\rho}_i^k(x) = \frac{-1}{N_i h^p} \sum_{x_j \in C_i} K(\frac{x - x_j}{h})$$

where $N_i = Card(C_i)$, $h$ is the sensibility (i.e. the width of kernel) and $K(x)$ is a kernel of probability.

A continuous $K(x)$ leads to a continuous estimated density. We used a kernel averaged per interval to thin the histogram estimation.

b/ Selection of the parameters :

Once the densities are estimated for each class and each parameter, the discrimination power of a parameter for a pair of classes is measured in terms of a recovering area. We use the following criterion :

$$\beta_{ij}^k = \int_E \text{Min}\left(\tilde{\rho}_i^k(x), \tilde{\rho}_j^k(x)\right) dx$$

The most discriminating parameter is the one that minimizes the recovering area :

$$u_{ij} = \text{ArgMin}_k(\beta_{ij}^k)$$

Notes:

* more robustness might be added by using convolved functions.

* selecting the best parameter for each pair of classes may not be a good idea because it sometimes leads to a large number of parameters, most of the times correlated. Two strategies can be used to reduce the set of selected parameters. The first is to apply a pre-filter over the set of parameters before using the discrimination method, for example a step-by-step selection method. The problem would then be that equivalent parameters could be selected. The second, which avoids this drawback, is to use the value of the test given by the method. We present the algorithm in the annex.

2. Unidimensional Gaussian Model.

This peculiar case of the general model leads to a very simplified version of the method.

a/ Density estimator

We denote by $\mu_i^k$ the mean of parameter $p_k$ for class $C_i$ and by $\sigma_i^k$ its standard deviation.

We suppose that the density function $\tilde{\rho}_i^k(x)$ of parameter $k$ for class $C_i$ is a gaussian one :

$$\tilde{\rho}_i^k(x) = \frac{1}{\sqrt{2\pi\sigma_i^k}} \text{ Exp } \left[\frac{-1}{2}\left(\frac{x - \mu_i^k}{\sigma_i^k}\right)^2\right]$$

The equation of the decision surface is :

$$\Pi_i C_{i/j} \tilde{\rho}_i^k(x) = \Pi_j C_{j/i} \tilde{\rho}_j^k(x)$$

with $\Pi_i$ = a priori probability of class $C_i$ and $C_{i/j}$ = cost of misclassificating a sample of class $C_i$ in class $C_j$.

We shall assume that all a priori probabilities are the same and that the costs of misclassification are functions of the standard deviations :

$$C_{i/j} \sim \sqrt{\sigma_i^k}$$

b/ Selection of the parameters :

This peculiar choice of the cost leads to a simplified test. The decision surface between class $C_i$ and class $C_j$ is then defined by :

$$(\frac{x - \mu_i^k}{\sigma_i^k})^2 = (\frac{x - \mu_j^k}{\sigma_j^k})^2$$

The solution inside interval $[\mu_i^k, \mu_j^k]$ is given by :

$$\frac{\mid \mu_i^k - \mu_j^k \mid}{\sigma_i^k + \sigma_j^k}$$

Thus, the discriminating power of parameter $k$ for pair $(C_i, C_j)$ is estimated by :

$$\beta_{ij}^k = \frac{\mid \mu_i^k - \mu_j^k \mid}{\sigma_i^k + \sigma_j^k}$$

finally, the most discriminating parameter is :

$$u_{ij} = \mathrm{ArgMax}_k(\beta_{ij}^k)$$

#### 4.3.3 Second Step : Decision.

1. The vote function.

   a/ General model

   We simply use a normalization of the pair of density functions to estimate the vote function of class $C_i$ versus class $C_j$ :

$$D_{ij}(x) = \frac{\rho_i^{u_{ij}}(x)}{\rho_i^{u_{ij}}(x) + \rho_j^{u_{ij}}(x)}$$

   and :

$$D_{ji}(x) = \frac{\rho_j^{u_{ij}}(x)}{\rho_i^{u_{ij}}(x) + \rho_j^{u_{ij}}(x)}$$

   where $x$ is the sample to classify.

   Note that it is necessary to apply a threshold when the density estimation at point $x$ is near zero. If needed, it is possible to change the a priori probability $\Pi_i$ or the cost $C_{i/j}$ simply by balancing this function. The vote function becomes :

$$D_{ij}(x) = \frac{\Pi_i C_{i/j} \rho_i^{u_{ij}}(x)}{\Pi_i C_{i/j} \rho_i^{u_{ij}}(x) + \Pi_i C_{i/j} \rho_j^{u_{ij}}(x)}$$

   b/ Gaussian model

   The same vote function can be used but we obtained better results with the well known fuzzy logic function displayed on diagram 10.

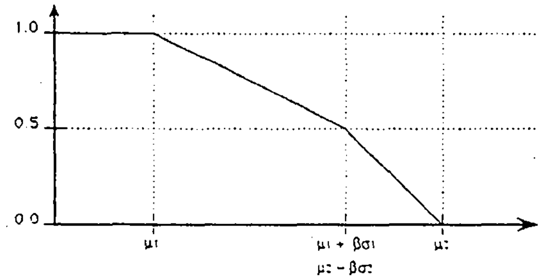


Figure 10: fuzzy logic function

2. The decision function : simple vote.

   To decide to which class a sample $x$ belongs, we build a $N \times N$ matrix $M$ where $m_{ij} = D_{ij}(x)$. $D_{ij}(x)$ represents, for sample $x$, the vote in favor of class $C_i$ for the pair $(C_i, C_j)$. It should be noted that each element of the matrix refers to a different set of parameters in parameters space. For each pair, we have selected the most favorable subspace for discrimination. A vote function is applied to this matrix to find the decision class. We implemented two well-known functions :

   a/ The fuzzy vote function :

   This function select the class which is globally the best.

$$F_d^1(x) = \text{ Arg Max}_i\,(\Pi_j D_{ij}(x))$$

   where :

$$V_i^1(x) = \Pi_j(D_{ij}(x))$$

   is the product of all the votes in favor of class $C_i$ versus all other classes $C_j$.

   Examples :

   Fuzzy case

$$M(x) = \begin{bmatrix} 1.00 & 0.70 & 0.80 \\ 0.30 & 1.00 & 0.05 \\ 0.20 & 0.95 & 1.00 \end{bmatrix}$$

$$F_d^1(x) = C_1$$

   Binary case

$$M(x) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$F_d^1(x) = C_1$$

   b/ The minimax function :

   This function, from games theory, minimizes the risk of misclassification.

$$F_d^2(x) = \text{ Arg Max}_i \text{ Min}_j\,(D_{ij}(x))$$

   we note :

$$V_i^2(x) = \mathrm{Min}_j(D_{ij}(x))$$

Examples:

Fuzzy case

$$M(x) = \begin{bmatrix} 1.00 & 0.70 & 0.80 \\ 0.30 & 1.00 & 0.05 \\ 0.20 & 0.95 & 1.00 \end{bmatrix}$$

$$F_d^1(x) = C_1$$

Remark :

The reader might have noticed the amazing fact that the opposition between classes $C_j$ and $C_k$ is used when class $C_i$ is decided. We have to do this because the result of each vote might influence the general decision (depending on the vote function). The reason for that appears with ambiguous votes. In the following example, each class have "lost" at least one time against another one :

$$M(x) = \begin{bmatrix} 1.00 & 0.51 & 0.51 \\ 0.49 & 1.00 & 0.99 \\ 0.49 & 0.01 & 1.00 \end{bmatrix}$$

The computations give :

$$F_d^1(x) = C_2$$

and :

$$F_d^2(x) = C_1$$

Here, the fuzzy logic function, that takes into account global information, appears to be the most natural choice.

3. Improvements the decision function.

We implemented the following tests to improve the reliability of the votes :

a/ Crossed Vote :

The vote is confirmed if :

$$F_d^1(x) = F_d^2(x)$$

Else, the sample is said to be uncertain.

b/ Vote with majority :

If the decision is class $C_i$, then $V_i(x)$ must be greater than a threshold $s$, which means that a minimum amount of confidence is required to accept the vote.

$$V_i(x) > s$$

c/ Vote with deviation :

The deviation between the two best votes must be greater than a threshold to be accepted. If we note $V_{max}$ the highest vote and $V_{next}$ the second highest vote, the test is :

$$\left(\frac{V_{next}}{V_{\max}} < s^1\right)$$

d/ Mixed decision

If all these tests lead to an indecision vote, the context often says us more than "I don't know", for instance it might mean :"it must be class one or class two". This is an indecision, but an indecision with knowledge. For this reason, we introduce now the mixed classes. A mixed class is an indecision class corresponding to the confusion between two classes. The condition is :

if : $\left(\left(\frac{V_{next}}{V_{\max}} > s_m^1\right) \wedge \left(V_{\max} + V_{next} > s_m^2\right)\right)$

then $x$ belongs to the mixed class $(C_{max}, C_{next})$.

In the gaussian model, another definition can be set. Around the fifty-fifty chance point, a mixed indecision class is defined. The mixed class becomes an interval.

The mixed indecisions will be solved later using for instance spatial constraints.

4. Comments and results.

a/ General case :

The general case does not lay on any model. It selects the best parameters (in the sense of the minimum recovering area) without any analysis of the density function. We point out that a great complexity of density functions does not change the complexity of the method. The most sensitive point is that the estimation of density is often delicate to perform when a large number of samples is not available. In our cases, the results are always correct because we have a large learning sample (thus a good density estimation) and a large set of parameters (thus a large choice). The recognition rate is good (90% to 98% depending on the difficulty).

b/ Gaussian Case :

This model is easy to implement and works fast. It is robust because the estimated parameters are robust. It gives good results but selects only gaussian-like parameters. If a selected parameter is not gaussian, the results of discrimination will be feeble. From our experience, it seems that the recognition rate between classes is well balanced.

Remarks :

In our method, the "curse of multi-dimensionality" is conjured by reducing the problem to a collection of smaller dimension spaces (see [Han81]). The fact that we work on pair of classes and the separation between discrimination and decision allows to :

- add a new class without invalidating the discrimination scheme : only the new pairs created have to be added (the new class with all already existing classes).

- work in a subspace of classes (by using only the pair of classes of this subspace).

- change easily the a priori probability and the cost of misclassification of each class (by changing the weights in the vote function)

- insert a new parameter in the system without invalidating the rest of it (just test if the new one can do better than the already selected ones for each pair)

All these features make the method a very adaptative one.

## 4.4 Results

We show some results on three libraries

1. **Brodatz textures**

We tried and classify the nine following textures from the well known Brodatz album : wood, canvas, water, grass, wool, ivy, sand, bubble and raffia. Several authors have reported good results on these textures, using all types of parameters. The training set was made of 64 images of size 64 x 64 per class (figure 11).

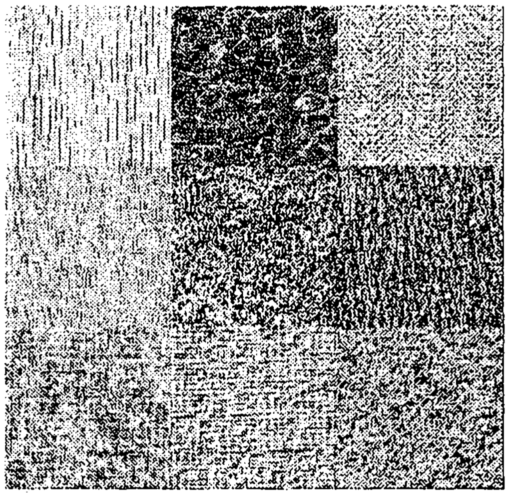

Figure 11: Brodatz textures

547 parameters were computed among all types, fractal, statistical, autoregressive, integral geometry. After data analysis, 8 parameters were selected, seven of Haralick and one run length. It must be noted that all parameters selected are co-occurrence ones with small translation length, which leads us to classify our library as a Markovian one.

The mean recognition on a 20 % test sample set is 97.77 %, which is better than all previously published results, to our knowledge.

In this case, we see that making some well chosen parameters cooperate in classification can improve significantly the results. Correlations between parameters and a multiplot are shown on figure 12.

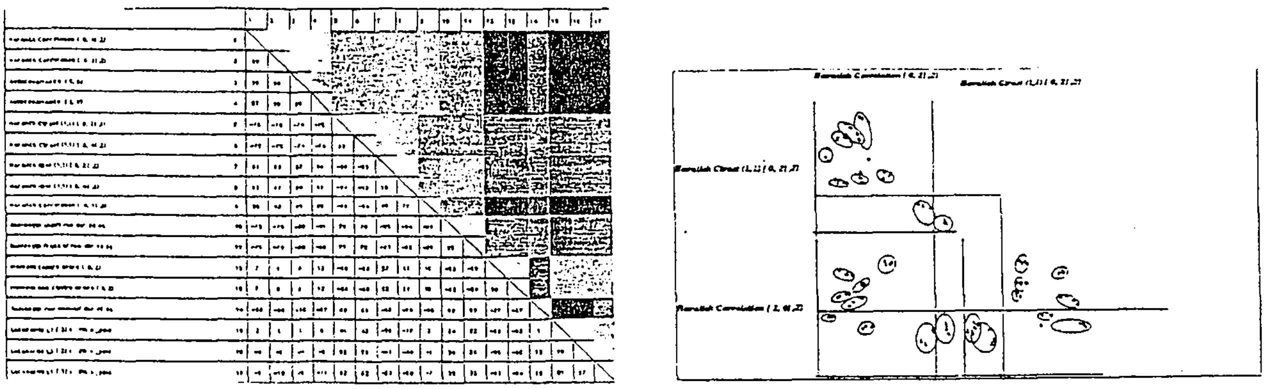

Figure 12: correlations and multiplot of parameters computed on Brodatz textures

2. **Binomial textures** :

A 2D binomial texture is obtained through the following process : the image is divided into four quarters, the first one receiving mass (intensity) $m_1$, the second $m_2$, the third $m_3$, the fourth $m_4$, with $m_1 + m_2 + m_3 + m_4 = 1$.

The process is then interated in each fourth, giving birth to subsquares of mass $m_i m_j$ depending on the place. We go on until pixel size is reached. The resulting texture is of course very irregular. density.

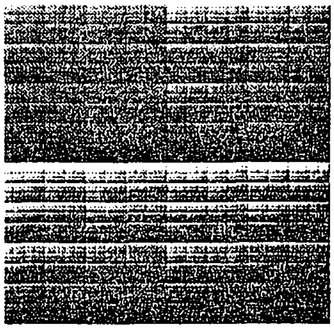

Figure 13: Binomial textures

The four classes on figure 13 are obtained with four different choices of $(m_1, m_2, m_3, m_4)$. The same parameters as earlier were computed and this time, three multifractal parameters were selected and able to achieve a 100% recognition rate. Of course, we classify our library as a fractal one.

3. **Cork texture**

This case is more difficult since we try here to separate four states of one texture rather than four different textures (figure 14).

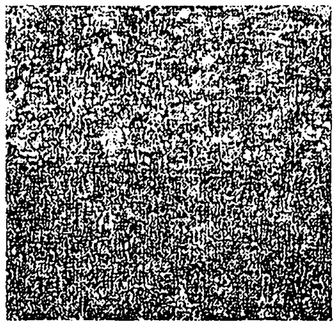

Figure 14: Cork textures

The variations of appearance of the cork pieces are due to their processing, where temperature and pressure cannot be totally controled. For the human eye, specially if you leave apart the grey levels mean which is subject to changes with ambient light, it is difficult to distinguish between the textures. Psychovisual tests have shown that non trained people do not achieve better than 70% classification rate. Using three parameters (both statistical and fractal), the system was able to reach 93% of good recognition with the same conditions as previously. There again, it is the method that provides us with a "model" in the sense that it chooses the best way to look at and measure our library, which we may classify as a mixed one.

# 5 Texture Segmentation

## 5.1 Introduction

The segmentation system, EXCALIBUR, works very much in the same way that ARTHUR, to which it is directly connected.
The same ideas of integration of different methods for global cooperation and advanced user interface have been applied.

Besides this integration, our contribution here lies in the possibility of making several methods cooperate at different levels and in the definition of mixed classes to refine the segmentation.

### 5.2 The segmentation process

The basic principle is very simple : the first thing to do is to define the set of images to be processed, then to choose an associated library from ARTHUR, meaning that EXCALIBUR will use a (small) set of parameters and a vote method connected to the data analysis module.

It is also possible to decide at which resolution the images will be processed, to save computing time. The non computed points will be affected using spatials constraints. Then, the system computes all parameter images (these are the transforms of the original image under the selected parameters) and, using the vote scheme, the decision image, which is a first approximation of the segmentation.

### 5.3 The post-processings

Most of the times, the segmented image has to be refined to obtain a good result. A lot of post-processings are possible in EXCALIBUR, we only mention a few of them, and the fact that a tool box is available where you can pick series of operations and very simply chain them using the graphic interface to enhance your original decision images. The system memorizes the serie of processings so that one may think of it as just one complex operation.

The processings are grouped into several categories :

1. **Qualitative**

   These are not strictly speaking processings, but rather tools designed for providing a qualitative insight of the segmentation: display of a specific parameter image, or of variations of one parameter along a direction of the image (see figure 15). Selective display of only one or a few of all the texture classes instead of the whole image.

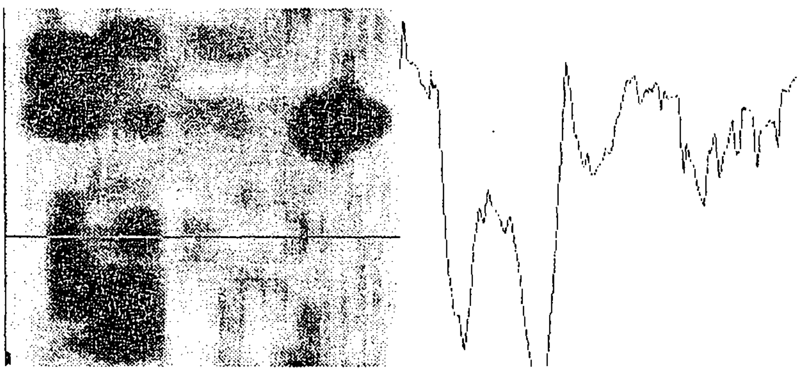

Figure 15: lacunarity image of Brodatz textures composite image -see original image below-, and variation of lacunarity along a line

2. **Morphological operators**

   These include all classical morphological operators, such as conditional erosion and dilation, openings, removing of small connected components, etc ...

3. **Geometrical operators**

These are geometric filters as for instance $K$-nearest neighbors filter or re-affectation of boundary zones.

4. **Context operators**

They allow to take into account some high level knowledge about the images : for instance, for a specific type of images, class A can never be a neighbor to class B, or class C is always included in class D.

5. **Mixed Classes**

All texture segmentation methods use, at a certain time, a window centered on the point to be classified, and do some statistical computations inside this window. If the considered point lies in the middle of a zone of pure texture, we may hope that the system will give a correct answer. But when the point is on or near a boundary, the measurement will have nothing to do with reality. The question will be : "Is that window included in texture A, B, or C ?". Since the window is in no pure textured zone, we shall always get a false answer. We do have here a paradox, because we are interested in the boundaries between textured zones, but we do not give the system any chance to behave correctly near these boundaries. Computing windows are generally 16 x 16 pixels wide, thus, in the best cases, we shall have an uncertainty of width 8 pixels around each boundary, if we do not take into account small regions of one texture included in another one and other special cases that also lead to false answers or uncertainties. If we want to refine our segmentation, we must certainly look at what is really hapenning in mixed zones. Our very simple idea is then to tackle the problem of the boundaries by including them into the system, meaning that we teach the system how to react when it meets a boundary zone. The first way would be to say that if texture A is statistically charaterized by feature vector U and texture B by vector V, then a computing on a boundary window would lead a feature vector W "statistically between U and V", to put it loosely. Unfortunately, this is not true, as it can be seen with the following simple example : take A to be a chess-board texture with alternate black and white blocks of a certain size, and let B be the same texture, only the chess-board is shifted in such a way that the white blocks of A become the blacks ones in B. The computation of lacunarity on A and B gives the same result, $L_A = L_B$, but on a boundary window, the lacunarity $L'$ will be different, and thus $L'$ is not included in $[L_A, L_B]$. A better method is, as usual, to let the system decide by itself what a mixed zone is.

Two methods have been implemented :

- Virtual Mixed Classes : during the learning of the textures by ARTHUR, we let the system consider virtual mixed classes by creating mixed windows with certain proportions of two pure classes, and computing all the parameters on these virtual classes. In the classification phase, these results are taken into account on the same level as those coming from the pure classes. In this method, if the inputs are textures A, B, and C, the system will create virtual classes A-B, A-C, and B-C, and will go into the whole process with these six classes. Another way to think of mixed windows is to see them as special classes, namely frontier textures between each pair of classes. The advantage of this method is that we need no assumptions or theoretical approximation, thus we do not reduce to any peculiar model, we just let the system find the best model in each case. The drawback is that the number of classified classes grows very fast, and if there are more than 6 pure classes, the method becomes impractical.

- Parametric Mixed Classes : this scheme has been presented in the Data Analysis section. Here, we do not create mixed windows, but directly mixed classes. We say that a specific point is in mixed class A-B if its probabilities to belong to classes A and B are both superior to a certain threshold (more precisely there is a possibility of confusion between A and B, and only between these two classes). In that case, we will not affect the point to class A nor to class B, and we will neither say that it is an uncertainty point, but rather a mixed class A-B point. The advantage of this method is that it is simpler than the previous one. The drawback is that it is founded on a assumption, related to the discriminant analysis scheme, and thus is sensitive to errors in the probability density functions estimation.

However, both methods give good results not only in localizing with better precision the boundaries, but also in reducing the overall rate of misclassification (see Results, next paragraph).

Remark : it is possible to consider mixed classes composed of more than two textures, for instance to detect triple junctions, but the computation cost then grows very quickly.

## 5.4 Results

We present three segmentation results :

1. **Brodatz textures**

We used here the library defined in section I.4.1 to study image figure shown on 16, where textures of wood, canvas, bubbles and water have been mixed.

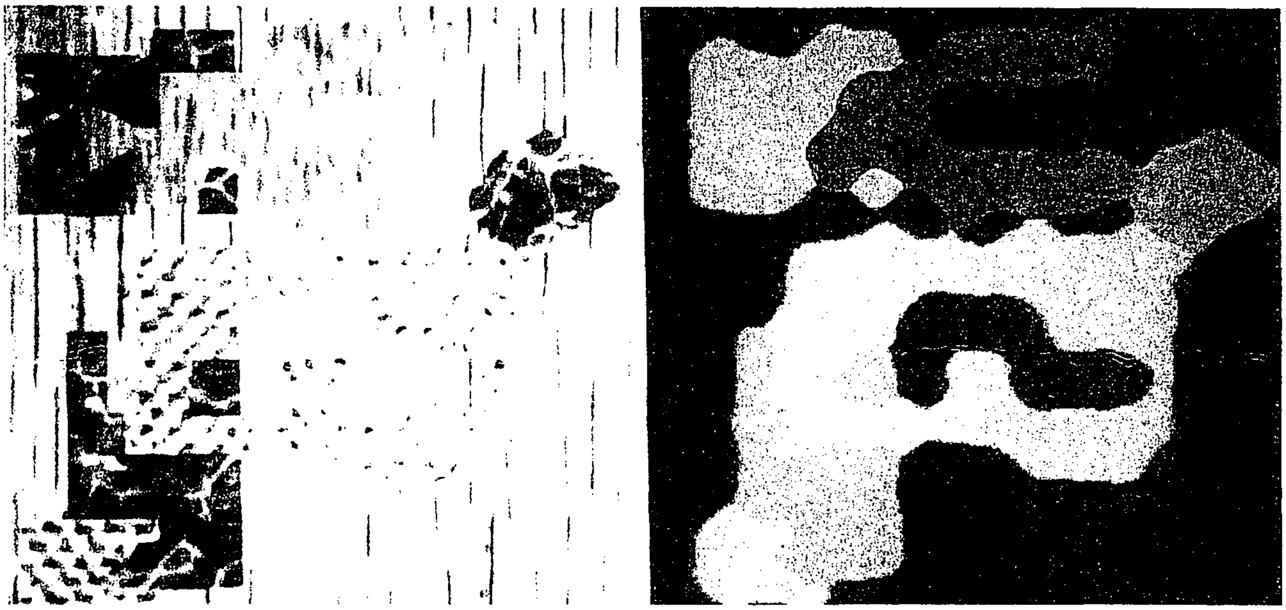

Figure 16: composite image of Brodatz textures and its segmentation

The difficulties of the original image are the thin line of wood between water and canvas, the small blob of bubble inside the water zone, the small blob of water inside the bubble zone, and some irregular boundaries between bubble, wood, and canvas. As one can see, most of the features of the original image have been captured by the segmentation, including the thin wooden line, the small blob of bubble and almost all irregular boundaries. The system however failed in detecting the small blob of water inside bubbles and had some trouble with the thin wooden line. The fact that small zones (the wooden line is 6 pixels wide and the bubble blob is an irregular shape that can be included inside a 12 x 12 pixels square) could be detected by EXCALIBUR is a result of the use of mixed classes. On figure 17 is displayed a preliminary segmentation using mixed classes, where the color of each detected mixed class is the interpolation of the colors of the two correspondent pure classes. The mixed classes appear at the boundaries between the corresponding pure classes, and this fact allows a better localization of the transitions. A simple processing in order to get the final segmentation image is then for instance to conditionally dilate in parallel all pure classes within the domain composed of the pure class itself, all the associated mixed classes, and the uncertainty points.

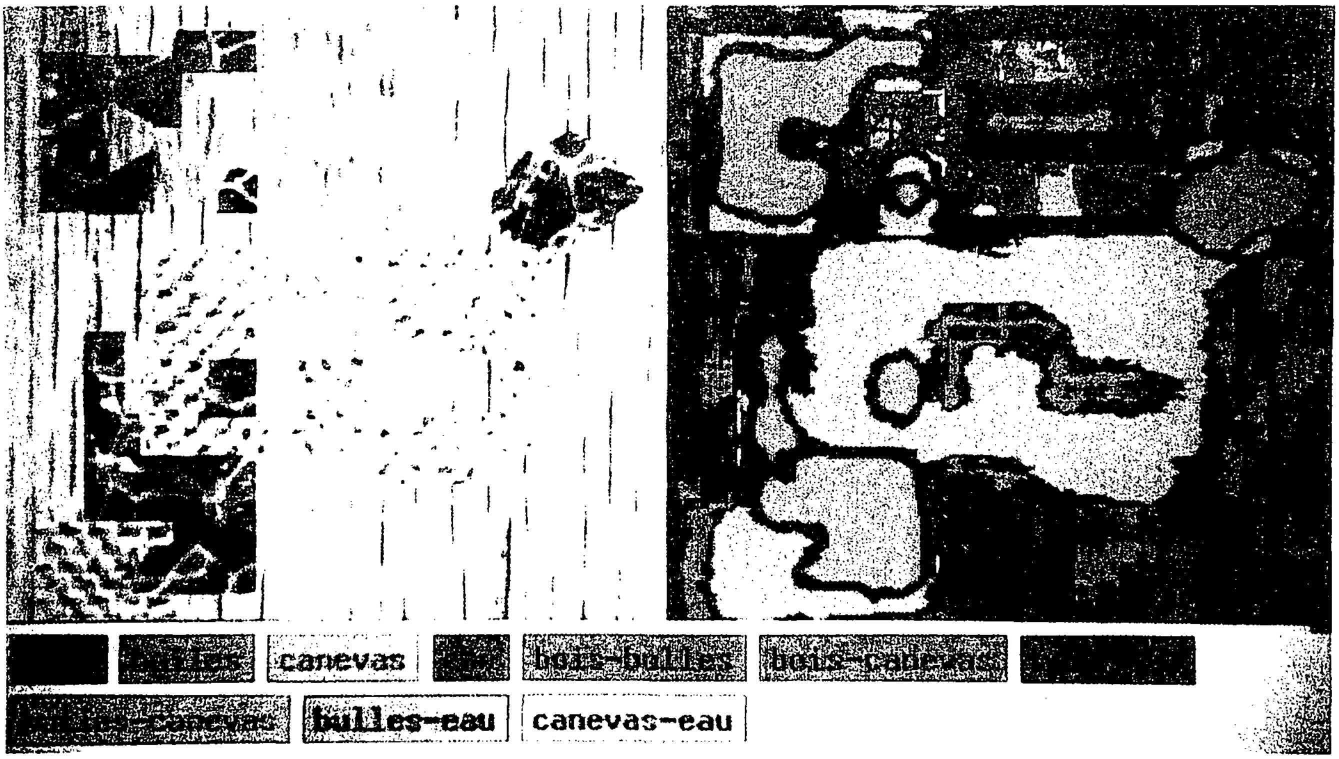


Figure 17: original image and first step of segmentation using mixed classes

2. **MR image of the eye**

The learning was made on 25 MR images of the eye, each class containing 50 samples.

In order to allow simple learning when no images of pure texture are available (which happens here and very often) ARTHUR is designed in such a way that it can take as an input parallelogram-like zones defined by three points in a composite image. The zones are then processed in order to make possible the computation of the parameters.

In our case, the samples were parallelogram-like zones of approximatively 12 x 12 pixels. The original eye and its segmentation are on figure 18.

Here again, EXCALIBUR was able to detect all main features of the image.

3. **MR image of the thorax**

The image on figure 19 is a difficult one since it is very noisy (it includes a large vertical strip of " pure " noise due to blood flow) and it contains several textured zones with irregular boundaries, most of which are very small : heart, lungs, bone, blood flow noise, exterior part. The learning was made under the same conditions as those of the eye image. As can be seen on figure 19, EXCALIBUR was able to localize correctly the lungs, the exterior part, the blood flow, most of the bone, but failed in detecting the heart which is indeed a very small part of the image whose texture is blurred by the noise and masked by other morphological structures.

It is worth noticing that classical edge detection methods are performing poorly on those images.

# 6 Conclusion

In this work, we have studied a new approach for image analysis, in which the multifractal theory seems to be of great help for segmentation and interpretation. We have also presented an unified system for texture segmentation that gives good results on complex real images, like medical ones. We think that an larger integration with other techniques and theories - as for instance Wavelets and Non Standard Analysis - could lead to a significant progress in automatic image comprehension.

Figure 18: MR image of the eye and its segmentation

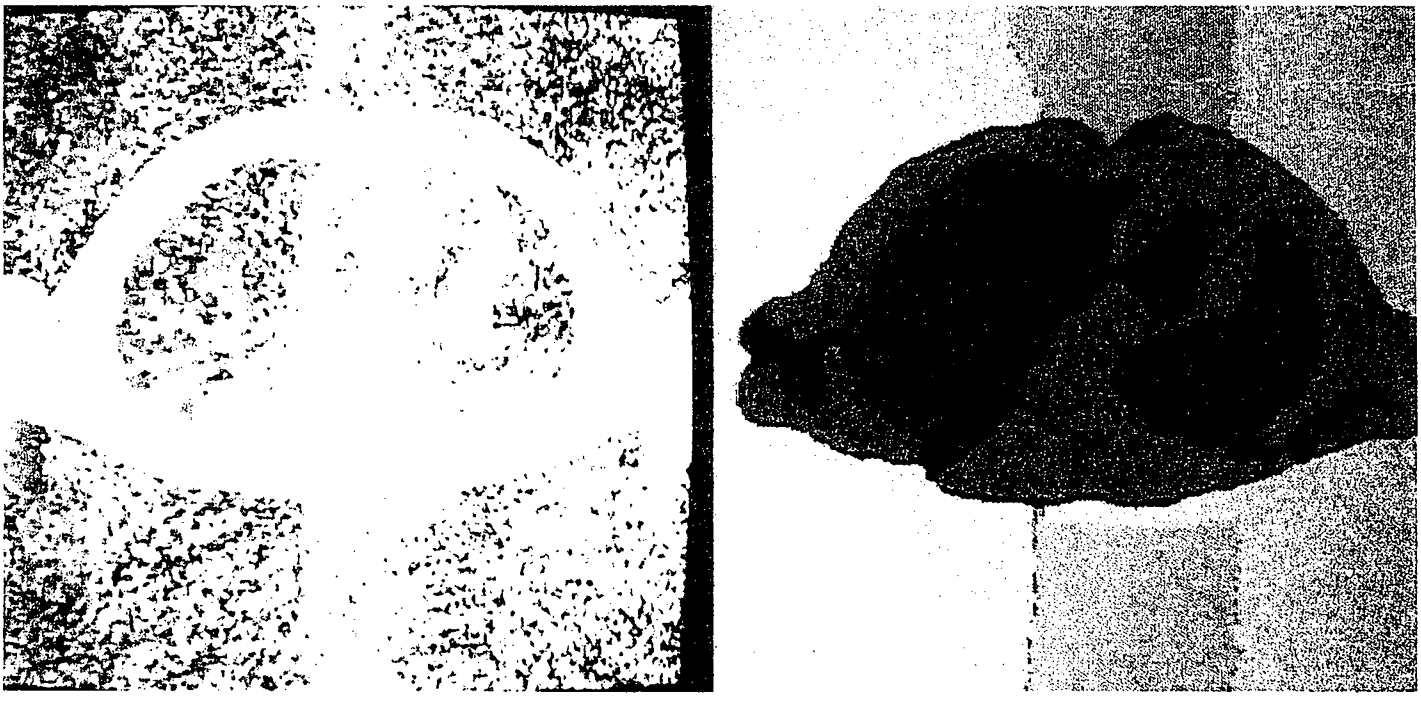

Figure 19: MR image of the thorax and its segmentation

## 7 Acknowledgements

The authors wishes to thank Yves Lechevallier for a lot of helpful clues about discriminant analysis. Many thanks also to Jean-Paul Chieze for computer support, and to Nathalie Gaudechoux.

## 8 Annex : Lowering of the number of parameters.

We present the method for a one-dimensional model but it can be easily extended to higher dimensions by the use of N-uplet.

For each pair, we define the two following sets :

$$E_{ij}^{>} = \left\{ P_k / \beta_{ij}^{k} > s^{>} \right\}$$

(set of "sufficiently" discriminating parameters)

$$E_{ij}^{\sim} = \left\{ P_k / \frac{\beta ij}{\mathrm{Max}_k \beta_{ij}^{k}} > s^{\sim} \right\}$$

(set of equivalent parameters)

An acceptable set of parameters for each pair is defined by:

$$E_{ij} = E_{ij}^{>} \cup E_{ij}^{\sim}$$

The set of all the discriminating parameters is :

$$E = \bigcup_{\substack{i \neq j \\ i < j}} E_{ij}$$

The problem is : find the minimum set of variables $E_{min}$ such that :

$$\forall E_{ij}, \quad E_{\min} \cap E_{ij} \neq \emptyset$$

This problem is a NP one.
We propose the following procedure to be done until the set $E$ is null:
Step 1: select the parameter

$$P_{\mu} = \mathrm{ArgMin}_{(i,j)} \mathrm{Max}_k \left( \mathrm{Card}(P_k) / P_k \in E_{ij} \wedge E_{ij} \in E \right)$$

where :

$$\mathrm{Card}\ (P_k) = \mathrm{card} \left\{ (C_i, C_j) / P_k \in E_{ij} \right\}$$

Step 2: update the set E

$$E \leftarrow \left\{ E \backslash\ E_{ij} / P_k \in E_{ij} \right\}$$

This procedure provides us with one of the minimal coverings of $E_{ij}$. The parameter assigned to a pair of class is the one that have the best test.

## References

[AKA88] Amar Ait-Kheddache and Sarah A.Rajala. Texture classification based on higher-order fractals. IEEE, 1988.

[AR81] Narendra Ahuja and Azriel Rosenfeld. Mosaic models for textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(1):1–11, January 1981.

[BCG90] Alan Conrad Bovik, Marianna Clark, and Wilson S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, January 1990.

[BD85] M. Barnsley and S. Demko. Iterated function system and the global construction of fractals. *Proceedings of the Royal Society*, A 399:243–245, 1985.

[B.D88] B.Dubuc. *Evaluating the Fractal Dimension of Surfaces.* PhD thesis, McGill University, July 1988.

[BEHL86] 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.

[B.M77] Benoit B.Mandelbrot. *The Fractal Geometry Of Nature.* W.H.Freeman and company, 1977.

[Ca90] G. Celeux and al. Analyse discriminante sur variables continues. *Collection Didactique INRIA*, 1990.

[CL82] G. Celeux and Y. Lechevallier. Méthodes de segmentation non paramétriques. *R.S.A.*, 30(4):39–53, 1982.

[FJ91] F. Farrokhina and A.K. Jain. A multi-channel filtering approach to texture segmentation. In *CVPR*, 1991.

[FK80] Olivier D. Faugeras and Pratt William K. Decorrelation methods of texture feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(4):323–332, July 1980.

[Gal75] Mary M. Galloway. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4:172–179, 1975. Edited by Academic Press.

[GG88] André Gagalowicz and Christine Graffigne. Blind texture segmentation. In *Proc. International Conference on Pattern Recognition.* IEEE, October 1988. Pékin, Chine.

[GM85] André Gagalowicz and S.D. Ma. Sequential synthesis of natural textures. *Computer Vision, Graphics and Image Processing*, 30, 1985.

[Han81] D.J. Hand. *Discrimination and Classification.* Wiley, 1981.

[Har79] Robert M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.

[HI83] H.G.E Hentschel and I.Procaccia. The infinite number of generalized dimensions of fractals and strange attractors. *Physica 8D*, 1983.

[Hut81] J. Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30:713–747, 1981.

[I.P] I.Procaccia. The characterization of fractal measures as interwoven sets of singularities : Global universality at the transition to chaos. Technical report, Department of Chemical Physics, The Weizmann Institute of Science, Rehovot 76100, Israel.

[Kah90] B. Kahng. Negative moments of current distribution in random resistor networks. *Physical Review Letters*, 64:914-917, 1990.

[KCC89] James M. Keller, Susan Chen, and Richard M. Crownover. Texture description and segmentation through fractal geometry. *Computer Graphics and Image Processing*, 45:150–166, 1989. Edited by Academic Press.

[LDG+90] S. Lovejoy, A. Davis, P. Gabriel, Schertzer D., and G. Austin. Discrete angle radiative transfer. 1. scaling and similarity, universality and diffusion. *Journal of Geophysical Research*, 95:699–715, 1990.

[LV88] Jacques Levy-Vehel. *Analyse et synthese d'objets bi-dimensionnels par des méthodes stochastiques*. PhD thesis, Université de Paris Sud, Decembre 1988.

[LV90] J. Levy Vehel. About lacunarity, some links between fractal and integral geometry, and an application to texture segmentation. In *ICCV*, 1990.

[LV91] J. Levy Vehel. Fractal probability functions : an application to image analysis. In *CVPR*, 1991.

[LVB91] P. Levy Vehel, J. Mignot and J.P. Berroir. Applications of multifractal theory and texture segmentation to image analysis. In *INRIA Internal Report*, 1991.

[LVJ91] Y. Lechevallier, Levy Vehel, and P. J., Mignot. Arthur : un systeme d'analyse de textures. In *RFIA*, 1991.

[Man] B.B. Mandelbrot. Fractal measures (their infinite moment sequences and dimensions) and multiplicative chaos: Early works and open problems. Technical report, Physics Department, IBM Research Center, Mathematics Deparment, Harvard University, Cambridge, MA 02138, USA.

[Man82] B.B Mandelbrot. *The Fractal Geometry of Nature*. CA: Freeman, San Fransisco, 1982.

[Man89] B.B. Mandelbrot. A class of multinomial multifractal measures with negative (latent) values for the dimension $f(\alpha)$. In *Fractals (Proceedings of the Erice meeting)*. L.Pietronero, New York, 1989.

[MC91] B.S. Manjunath and R. Chellapa. A computational approach to boundary detection. In *CVPR*, 1991.

[Oon89] Y. Oono. Large deviation and statistical physics. *Progress of Theoritical Physics Supplement*, 99:165–205, 1989.

[Pen84] A. Pentland. Fractal-based description of natural scenes. *IEEE*, PAMI-6, 1984.

[PIP91] R.W. Picard, Elfadel I.M., and A.P. Pentland. Markov/gibbs texture modeling: Aura matrix and temperature effects. In *CVPR*, 1991.

[PNHA84] S. Peleg, J. Naor, R. Hartley, and D. Avnir. Multiple resolution texture analysis and classification. *IEEE*, PAMI-6(4), July 1984.

[PS88] H.O. Peitgen and D. Saupe. The science of fractal images. In *Springer Verlag*, 1988. New York.

[San76] Luis A. Santalo. *Integral Geometry and Geometric Probability*, chapter Integral Geometry in the Plane. Addison Wesley, 1976.

[Shr86] T.C. Hasley M.H. Jensen L.P. Kadanoff I. Procaccia B.I. Shraiman. Fractal measures and their singularities: The characterization of strange sets. *Physical Review*, February 1986.

[Sul] John R. Sullins. Distributed learning of texture classification.

[VP87] Harry Voorhees and Tomaso Poggio. Detecting textons and textures boundaries in natural images. *Proc. of IEEE*, pages 250–258, 1987.

[WDR76] Joans S. Weszka, Charles R. Dyer, and Azriel Rosenfeld. A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(4):269–285, April 1976.



Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique.

ISSN 0249-6399