

ARTHUR : Un système d'analyse de texture

ARTHUR : A System for Texture Analysis



Pascal MIGNOT

INRIA-Rocquencourt
Domaine de Voluceau
Rocquencourt, BP 105
78153 Le Chesnay Cedex, France

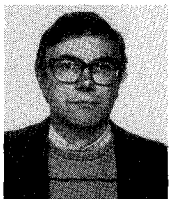
Pascal Mignot est diplômé de l'ESIEA (1991) et a obtenu un D.E.A. d'analyse d'image en 1992. Il effectue actuellement son service militaire, avant de démarrer une thèse en analyse d'images fin 93.



Jacques LEVY VEHEL

INRIA-Rocquencourt
Domaine de Voluceau
Rocquencourt, BP 105
78153 Le Chesnay Cedex, France
e-mail : jlv@bora.inria.fr

Jacques Lévy Véhel a obtenu le diplôme de l'École Polytechnique en 1984 et celui de l'École Nationale Supérieure des Télécommunications en 1986. Il est actuellement chargé de recherche à l'INRIA, et travaille dans le domaine de la géométrie fractale, appliquée à l'analyse de signaux, la prédiction et la modélisation.



Yves LECHEVALLIER

INRIA-Rocquencourt
Domaine de Voluceau
Rocquencourt, BP 105
78153 Le Chesnay Cedex, France

Yves Lechevallier a obtenu le diplôme de l'ISUP en 1974 et sa thèse concerne les méthodes de classification automatique. Il est actuellement directeur de recherche à l'INRIA-Rocquencourt. Ses domaines de recherches actuels s'inscrivent dans les méthodes de discrimination par arbre de décision et des méthodes de classification.

RÉSUMÉ

On présente un système complet pour la classification de textures, fondé sur une approche apprentissage-reconnaissance et qui intègre plus de deux mille paramètres. Ce système est totalement interactif et ouvert en ce sens qu'il permet l'intégration facile de nouveaux paramètres. Il inclut un outil puissant d'analyse de données, qui autorise le mélange de diverses

familles de paramètres et produit un arbre de décision qui donne, dans tous les cas testés, des résultats meilleurs que ceux publiés jusqu'ici.

MOTS CLÉS

Textures, co-occurrences, fractals, analyse de données, classification.

ABSTRACT

A complete system for texture classification is presented. It is based on a training approach, and is able to compute over two thousand parameters. It is a totally interactive and open system which allows easy integration of new parameters. A powerful data analysis tool is included. It permits the

merging of several types of parameters and builds a decision tree that generally gives better results than previously published ones.

KEY WORDS

Textures, co-occurrences, fractals, data analysis, classification.

1. Introduction

La littérature sur l'analyse de texture est très volumineuse, et depuis les articles classiques d'Haralick, de nombreux auteurs ont proposé de nouveaux paramètres fondés sur telle ou telle propriété des textures ([15], [34], [32], [1], [18], [28], [31], [9], [3], [12], [17], [11], [19], [20]). Parmi les classes les plus importantes, citons les co-occurrences de niveaux de gris, les dépendances spatiales, les paramètres fondés sur l'analyse de Fourier, sur la géométrie fractale, ou sur l'utilisation d'un modèle régressif, ou encore l'approche par textons.

Chacune de ces classes de paramètres fournit des résultats satisfaisants, voire excellents, mais seulement sur des types bien déterminés de textures. Certains articles (par exemple [34]) donnent des comparaisons détaillées entre certaines familles pour certains ensembles de textures. Cependant, aucune étude générale n'a été menée. D'autre part, même si des travaux systématiques d'apprentissage des meilleurs paramètres pour un problème donné ont été publiés (en particulier le travail original de [31]), ils restent fragmentaires, en particulier en ce qui concerne la variété des paramètres possibles.

A la lumière de toutes les études menées, il semble clair qu'il n'existe pas de paramètres universels dans le sens où ils permettraient de classer toutes les textures possibles. Avec ARTHUR, nous proposons un système :

- qui intègre pratiquement tous les types de paramètres proposés dans la littérature, plus quelques nouveaux critères,
- qui autorise le calcul simultané de ces paramètres sur une bibliothèque de textures définie à l'avance,
- qui permet de comparer les performances de ces paramètres,
- qui donne, pour une bibliothèque fixée, l'ensemble des paramètres (en général en petit nombre, autour de 5) fournissant la meilleure discrimination possible.

2. Arthur

2.1. IDÉES GÉNÉRALES

Les idées de base qui ont présidé à la conception d'ARTHUR sont les suivantes :

- approche apprentissage-reconnaissance,
- analyse multi-paramétrique non hiérarchique,
- notion de bibliothèque de textures,
- utilisation d'une famille de méthodes de discrimination,
- description quantitative et qualitative des paramètres et des textures,
- convivialité poussée.

2.2. PRÉCISIONS

2.2.1. Apprentissage-Reconnaissance

Constater qu'il n'existe pas de paramètre universel, c'est dire que chaque problème particulier nécessitera sa solution et conduit naturellement à adopter un mécanisme où le système apprend dans un premier temps les différentes textures à classer, puis fournit une solution adaptée à ce problème seulement.

2.2.2. Analyse multi-paramétrique

Un des buts principaux d'ARTHUR est de pouvoir utiliser conjointement les capacités discriminantes de paramètres aussi différents que l'énergie au sens d'Haralick ou la dimension fractale. A la demande de l'utilisateur, le système calcule les paramètres sur toutes les textures, aussi la méthode de discrimination doit être opérante sur un nombre très grand de paramètres.

2.2.3. Notion de bibliothèque

Tous les résultats obtenus par ARTHUR (classification, arbre de décision, comparaison des paramètres, etc...) sont relatifs à une bibliothèque, et donc de validité limitée. Cependant, rien n'interdit de comparer des bibliothèques entre elles, en particulier par le truchement des résultats d'ARTHUR : finalement, ARTHUR ne sert pas seulement à classer des textures dans une bibliothèque, et à évaluer et comparer des paramètres, mais il permet aussi d'établir l'existence de grands types de textures, comme par exemple les textures Markoviennes, fractales ou mixtes, en fonction par exemple des meilleurs paramètres qui les discriminent.

Quatre niveaux existent :

1. la bibliothèque,
2. la classe de texture,
3. l'échantillon ou photo,
4. la vignette ou sous-image, de taille paramétrable, sur laquelle les calculs sont faits.

2.2.4. Systèmes d'analyse de données

Actuellement le module utilisé pour la discrimination est le programme DNP (Discrimination Non Paramétrique [7]), extrait du logiciel d'analyse de données SICLA [6] développé à l'INRIA. DNP effectue une segmentation non paramétrique.

2.2.5. Description quantitative et qualitative

ARTHUR fournit une description quantitative complète : pour chaque échantillon de texture, la valeur de tous les paramètres, toutes les moyennes et variances à tous les niveaux, les tests de Fisher de chaque paramètre, les valeurs des corrélations entre tous les paramètres, et enfin l'arbre de décision permettant de classer avec précision les textures. Pour chaque bibliothèque, chaque feuille de l'arbre de décision donne deux probabilités : la probabilité que l'échantillon soit dans telle classe, et la probabilité de trouver un élément d'une autre classe que la classe spécifiée (pureté du segment).

La description qualitative consiste en :

- la représentation multigraphe des résultats, suivant les paramètres choisis et à tous les niveaux (vignette, photo, classe),
- l'affichage des corrélations sous forme d'image de niveaux de gris,
- l'affichage des paramètres les plus pertinents pour une bibliothèque donnée (test de Fisher supérieur à un seuil),
- l'affichage des paramètres les plus corrélés ou les moins corrélés à un paramètre donné.

Ainsi ARTHUR permet des comparaisons qualitatives entre paramètres aussi bien qu'entre bibliothèques.

2.2.6. Convivialité

ARTHUR a été développé à partir de l'interface Athena-Widget. Tout a été fait de telle sorte qu'un utilisateur non spécialiste soit à l'aise : toutes les opérations peuvent se faire en cliquant avec la souris, les affichages graphiques sont nombreux, l'utilisateur peut à chaque instant changer d'avis, d'interrompre des calculs, revenir en arrière, rajouter ou enlever des paramètres ou des textures, etc...

Le déroulement des opérations dans ARTHUR est le suivant :

Phase 1 : On définit la bibliothèque par les différentes classes de textures et par les images correspondant à chaque classe. On choisit parmi tous les paramètres proposés ceux que l'on désire calculer. On décide de la taille des vignettes sur lesquelles les calculs seront effectivement faits, de la discrétisation des niveaux de gris, et de divers prétraitements (normalisation statistique ou géométrique, égalisation d'histogramme, etc...).

Phase 2 : ARTHUR effectue tous les calculs.

Phase 3 : On visualise, si on le désire, les multigraphes, les corrélations, les tests de Fisher.

Phase 4 : On réalise la discrimination, les quelques paramètres permettant la classification sont choisis et l'arbre de décision est fourni.

Phase 5 : Une nouvelle image est analysée suivant l'arbre et classée dans la bibliothèque.

2.3. LES PARAMÈTRES

Parmi les paramètres, citons sans être exhaustif :

2.3.1. Les co-occurrences de niveaux de gris et dépendance spatiale

Il s'agit principalement des paramètres classiques d'Haralick [15], des run-lengths [12], et des paramètres d'autocovariance [13].

L'utilisateur choisit les tailles de fenêtre de calcul, les paramètres spécifiques (énergie, corrélation, etc.), et s'il y a lieu, la direction et la symétrie.

2.3.2. Les paramètres fractals

ARTHUR peut calculer la dimension fractale, la dimension d'information, la dimension de corrélation, la lacunarité (ℓ), la lacunarité centrée (ℓ_c), le rapport des deux

(ℓ_c/ℓ) pour différentes tailles de fenêtres et avec divers seuils, plus les paramètres multifractals suivants ([22], [33], [2], [29], [16], [14], [23], [24], [8], [25]) :

- tous les D_q (dimension généralisée), définis par :

$$D_q = \lim_{\ell \rightarrow 0} \frac{\sum \log p_i^q}{\log 1/\ell} \times \frac{1}{q-1}$$

où p_i est la somme des niveaux de gris dans une boîte b_i de taille ℓ . D_0 est la dimension fractale du support de la mesure. Si q est positif (resp. négatif), $D(q)$ reflète la dimension de l'ensemble des points où la mesure est dense (resp. éparsé) et ceci d'autant plus que q sera grand. Les D_q peuvent être considéré des « moments » fractals.

- α_{\min} , α_{\max} , $f(\alpha^*)$ où :

$$\alpha_i = \lim_{\ell \rightarrow 0} \frac{\log p_i}{\log 1/\ell}$$

avec les mêmes notations que plus haut.

α_i mesure la singularité locale

$f(\alpha)$ est la dimension fractale de l'ensemble des points ayant même α .

α_{\min} (resp. α_{\max}) est le plus petit (resp. grand) des α_i

α^* est la valeur maximisant f .

L'ensemble des points où α est proche de deux est l'ensemble des points réguliers de la texture.

2.3.3. Les paramètres de géométrie intégrale

La longueur de Favard est définie par :

$$\ell = \int n(G) dG$$

où $G = dp d\Phi$ est une densité de droites sur le plan,

$n(G)$ compte le nombre de points d'intersection de la droite et de la texture.

Pour une courbe rectifiable, ℓ est égal à la longueur de la courbe. On généralise [19] en calculant :

$$\forall m \in \mathbb{N}, \quad \ell(m) = \iint \max(m, n(G)) dG.$$

2.3.4. Les moments

ARTHUR calcule les moments, les moments centrés, les moments invariants [13].

2.3.5. Les paramètres fréquentiels

ARTHUR calcule tous les paramètres classiques utilisant la transformation de Fourier [13], [34].

2.4. MÉTHODE DE DISCRIMINATION

L'objectif est de construire des règles de décision permettant de bien reconnaître les classes de textures. La conception d'ARTHUR favorise et incite l'utilisateur à choisir un

nombre important de paramètres. Aussi la méthode de discrimination doit jouer un rôle explicatif et d'aide à la sélection des paramètres. En rendant prioritaire ce rôle nous avons éliminé les méthodes de discrimination de type « boîte noire » et le choix s'est porté sur les méthodes de segmentation construisant des fonctions de décision binaires. L'obtention d'un arbre binaire produit une représentation hiérarchique des zones de l'espace des données et les fonctions de décisions binaires associées peuvent être écrites sous la forme de règles de production.

Dans la phase de reconnaissance la clarté et la simplicité de cette représentation rend possible un dialogue et donne à l'utilisateur une interprétation du pouvoir discriminant de chaque paramètre.

Le développement de ces méthodes a été amorcé par Morgan et Sonquist [27]. Le livre de Breiman, Friedman, Olsen et Stone [4] est une référence importante ainsi que l'article de Quinlan [30] dans le domaine de l'apprentissage. L'algorithme associé à ces méthodes est très simple ; il s'agit de construire de manière récursive un arbre binaire en sélectionnant à chaque nœud de cet arbre la meilleure question binaire. Les outils nécessaires à la construction d'un arbre de décision sont :

- la définition d'un ensemble de questions binaires,
- un critère d'évaluation,
- une règle d'arrêt de la construction de l'arbre,
- une règle d'affectation de chaque segment terminal à une classe a priori.

Les méthodes de segmentation ont deux représentations principales :

- l'une sous forme d'un arbre binaire, par exemple :

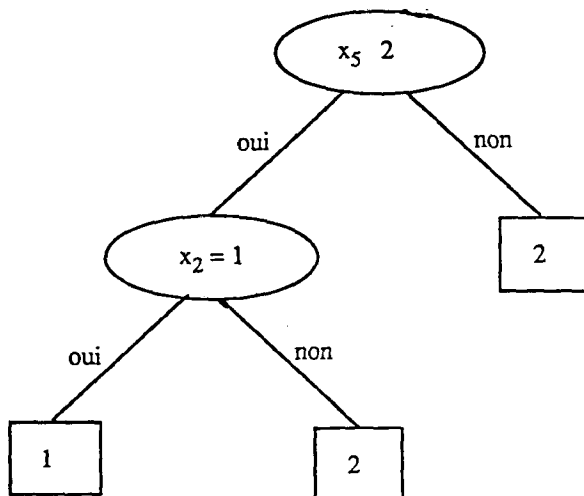


Figure 1. — Arbre de décision.

Cet arbre binaire est constitué, de deux questions binaires $(x_5 \leq 2)$? et $(x_2 = 1)$?, de trois segments terminaux. Un de ces segments est associé à la classe a priori numéro 1 et le deux autres à la classe a priori 2.

- l'autre sous la forme de règles de production, par exemple :

```

SI      (x5 ≤ 2)
ET      (x2 = 1)

ALORS  Classe a priori 1

SI      (x5 ≥ 2)
OU      ((x5 ≤ 2) ET (x2 ≠ 1))

ALORS  Classe a priori 2
    
```

2.4.1. Choix des questions binaires et de la règle d'affectation

Les paramètres calculés sur les textures sont des variables continues. Ainsi l'ensemble des questions binaires est l'ensemble de toutes les coupures possibles sur les variables continues sélectionnées. Nous avons choisi comme règle d'affectation de chaque segment terminal à une classe a priori la règle bayésienne dont le principe est le suivant : on note π_j les probabilités a priori associées aux k classes a priori et $C(i/j)$ le coût de mauvais classement d'une observation de la classe a priori j dans la classe d'affectation i . Pour une observation, représentée par ses p valeurs $x = (x_1, \dots, x_p)$ et par son affectation à la classe a priori j (noté $Y(x) = j$) nous décidons son affectation à la classe a priori i en utilisant la fonction de décision d (noté $d(x) = i$). Le coût de mauvais classement de cette fonction est égal à :

$$R(d) = \sum_{j=1}^k \pi_j \left[\sum_{i \neq j} C(i/j) P[d(x) = i / Y(x) = j] \right].$$

La fonction de décision d est appelée règle de Bayes si son coût est le plus petit possible.

2.4.2. Choix du critère d'évaluation

Chaque question binaire $(X \leq c)$? impose le découpage de la variable X sélectionnée en deux demi-droites représentant chacune une classe a priori. Dans ce cadre on veut choisir la coupure c de manière à rendre minimum le coût R de mauvais classement. Ce coût R de la fonction de décision d_c associé à cette question binaire $(X \leq c)$? s'écrit ainsi :

$$R(c) = \pi_1 C(2/1) P[d_c(x) = 2 / Y(x) = 1] + \pi_2 C(1/2) P[d_c(x) = 1 / Y(x) = 2].$$

Comme

$$P[d_c(x) = 2 / Y(x) = 1] = P[x \in]-\infty, c[/ Y(x) = 1] = F_1(c)$$

avec F_1 étant la fonction de répartition de la classe a priori 1 pour la variable X sélectionnée. D'où

$$R(c) = \pi_1 C(2/1) F_1(c) + \pi_2 C(1/2)(1 - F_2(c))$$

si on suppose que

$$\pi_1 C(2/1) = \pi_2 C(1/2) = \alpha$$

alors on a :

$$R(c) = \alpha(1 + F_1(c) - F_2(c)).$$

On remarquera que le $\text{Min}_c R(c)$ est équivalent au $\text{Max}_c |F_1(c) - F_2(c)|$ qui est la distance Kolmogorov-Smirnov (notée $D(c)$) entre les deux classes a priori. On choisit la région discriminante $\mathcal{R}_1 =]-\infty, c]$ si $F_1(c) > F_2(c)$, sinon $\mathcal{R}_1 =]c, +\infty[$. On estime $D(c)$ par

$$\hat{D}(\hat{c}) = \text{Sup}_x |\hat{F}_1(x) - \hat{F}_2(x)|$$

où \hat{F}_i sont les estimations des fonctions de répartitions F_i (Friedman [10], Celeux et Lechevallier [7]).

Lorsque le nombre de classes a priori est supérieur à 2 la première solution est de considérer la discrimination de k classes a priori comme une succession de discriminations de deux classes, en opposant chaque classe a priori aux autres. Mais ceci engendre k arbres de décision et la définition d'une règle majoritaire difficilement utilisable. Dans Celeux et Lechevallier [7] est proposé une autre solution qui permet de ne construire qu'un seul arbre. Il s'agit pour chaque variable et pour chaque coupure de rechercher le meilleur regroupement A des classes a priori. La variable retenue sera évidemment celle qui fournit le meilleur $\hat{D}(\hat{c})$. D'où

$$\hat{D}(\hat{c}) = \sup_x \sup_{A \in \mathcal{A}} |\hat{F}_A(x) - \hat{F}_{\bar{A}}(x)|$$

où \bar{A} est le complément de A , \mathcal{A} est l'ensemble de tous les regroupements possibles des classes a priori et \hat{F}_A est l'estimation de la fonction de répartition de A par :

$$\hat{F}_A(x) = \frac{1}{\pi_A} \sum_{i \in A} \pi_i \hat{F}_i(x)$$

avec

$$\pi_A = \sum_{i \in A} \pi_i.$$

On montre que la solution appartient à un sous-ensemble de \mathcal{A} de $(k-1)$ éléments ce qui annule, dans ce cas, le caractère combinatoire de la recherche.

2.4.3. Estimation du coût de mauvais classement

Ayant adopté cette stratégie il faut estimer le coût de mauvais classement associé à cette méthode.

Le critère d'évaluation utilisé est la distance de Kolmogorov-Smirnov. Ce choix entraîne que le coût de mauvais classement décroît à chaque segmentation de notre population. Donc l'estimation naturelle de ce taux, qui consiste à appliquer cette règle de décision à l'échantillon d'apprentissage, est une estimation très optimiste. Ce taux doit être calculé à partir d'un échantillon indépendant de l'ensemble d'apprentissage.

La stratégie adoptée est de tirer au hasard de l'ensemble des observations un sous-ensemble représentant 20 % de la population. Ce sous-ensemble servira d'ensemble test et les observations restantes constitueront l'ensemble d'apprentissage. L'avantage de cette technique est l'absence de biais dans l'estimation du taux d'erreur et sa facilité de mise en œuvre. D'autres possibilités d'estimer ce taux d'erreur sont décrites et proposées dans Breiman *et al.* [4], Mingers [26], et dans le livre « Analyse discriminante sur variables continues » [5].

3. Applications

Nous présentons des applications sur 3 bibliothèques :

3.1. TEXTURES DE BRODATZ

Cette bibliothèque est composée des neuf textures suivantes (voir *fig. 2*) : bois, bulles, canevas, eau, herbe, laine, lierre, raffia, sable.

Divers auteurs ont étudié ces textures et ont fait état de résultats satisfaisants avec divers paramètres (co-occurrences, run-lengths, etc...), atteignant jusqu'à plus de 80 % de taux de bonnes reconnaissances [12] [15] [31] [34].

Nous avons fait l'apprentissage sur 4 photos 256×256 de chaque classe, chaque photo étant découpée en vignettes 64×64 , ce qui donne 16 échantillons par photo, et 64 par classe. Une normalisation statistique a été effectuée sur chaque image.

547 paramètres ont été sélectionnés :

- 160 paramètres fractals : lacunarité, dimension,...
- 240 paramètres d'Haralick : l'énergie, l'entropie, la probabilité maximale, la corrélation, le moment aux différences inverses d'exposant 1, le contraste d'exposant 1 pour toutes les translations de taille inférieure ou égale à 4.
- 84 paramètres d'autocovariance, correspondant à toutes les translations de taille inférieure ou égale à 6.
- 20 paramètres de run-length soit les moments des traits courts, les moments de traits longs, la non-uniformité des niveaux de gris, la non-uniformité des traits, la fraction de l'image en trait, dans les 4 directions 0 degrés, 45 degrés, 90 degrés et 135 degrés.
- 43 paramètres de moments.

Sur la figure 3 est présenté l'arbre de décision après analyse des données : on voit que 8 paramètres ont été suffisants pour classer les textures.

L'arbre conduit aux remarques suivantes :

- Le taux de bonne classification est excellent sur la base de données, puisque les puretés de chaque segment et les probabilités de chaque classe dans chaque segment sont toutes de 100 %.

Pour valider ces résultats, nous avons fait des essais en utilisant une partie seulement de la base de données (80 %) comme échantillon d'apprentissage, et en effectuant la reconnaissance sur les images restantes. Les résultats des

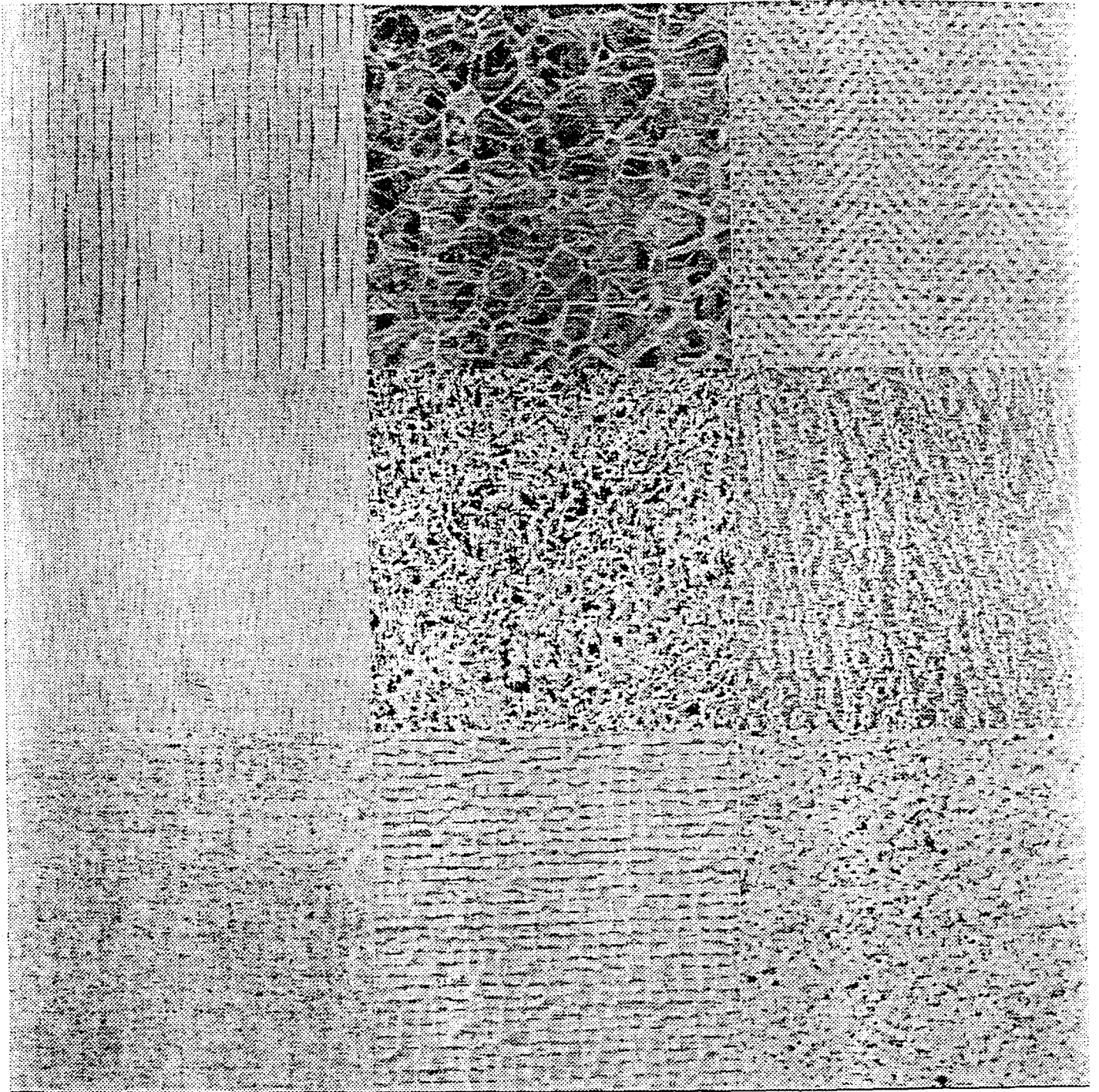


Figure 2. — 9 textures de Brodatz.

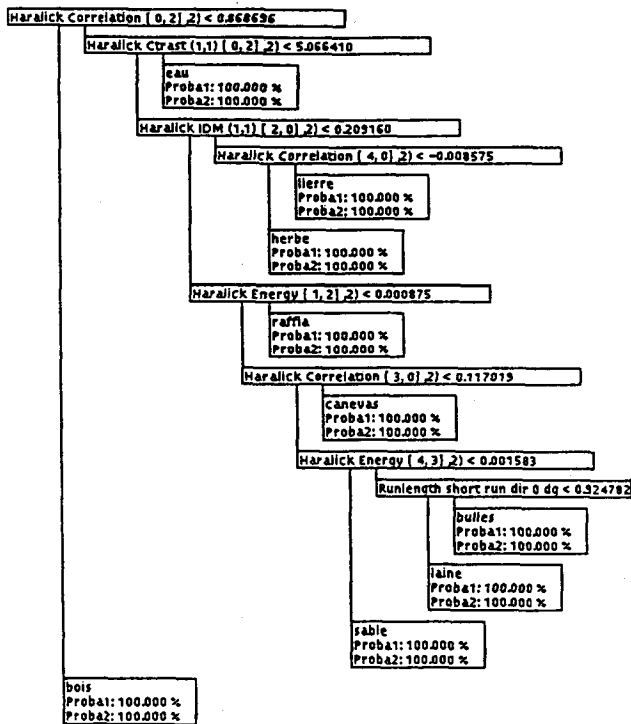


Figure 3. — Arbre de décision pour les textures de Brodatz.

cinq essais avec 20 % d'observations dans l'ensemble test sont les suivants :

- Le pourcentage de biens classés dans les cinq ensembles d'apprentissage est de 100 %.
- Le pourcentage moyen de bon classement des cinq ensembles test est de 97.77 % (il y a 12 mal classés sur 575).

| Classes d'affectation | Classes a priori | | | | | | | | | Total |
|-----------------------|------------------|--------|---------|-----|-------|-------|--------|--------|-------|-------|
| | bois | bulles | canevas | eau | herbe | laine | lierre | raffia | sable | |
| bois | 72 | | | | | | | | | 11 |
| bulles | | 55 | | | | | | 3 | 2 | 9 |
| canevas | | | 68 | | | | | | | 10 |
| eau | 3 | | | 57 | | 1 | | | | 14 |
| herbe | | | | | 63 | | 1 | | | 16 |
| laine | | 1 | | 1 | | 72 | | | | 21 |
| lierre | | | | | | | 59 | | | 11 |
| raffia | | | | | | | | 56 | | 11 |
| sable | | | | | | | | | 60 | 12 |
| Total | 75 | 56 | 68 | 58 | 64 | 73 | 60 | 59 | 62 | 575 |

Figure 4. — Tableau de confusion des cinq ensembles test.

Pourcentages de bon classement par classe a priori :

- bois 96,0
- bulles 98,2
- canevas 100,0
- eau 98,3
- herbe 98,4
- laine 98,6
- lierre 98,3
- raffia 94,9
- sable 96,8

• Les paramètres sélectionnés sont tous des paramètres de co-occurrence de niveaux avec une distance assez faible, ce qui tend à nous faire classer cette bibliothèque parmi les bibliothèques Markovienne (i.e. dont les textures peuvent être analysées à l'aide de paramètres markoviens).

Sur la figure 5 sont présentées les corrélations entre 17 paramètres, et sur la figure 6 un multigraphe à 3 paramètres.

3.2. TEXTURES BINOMIALES

Cette bibliothèque est composée de 4 textures binomiales (fig. 7) obtenues de la façon suivante.

On coupe l'image en quatre quarts, le premier quart reçoit la « masse » (intensité) m_1 , le deuxième m_2 , le troisième m_3 , le quatrième m_4 , avec $m_1 + m_2 + m_3 + m_4 = 1$. On itère ensuite le processus en divisant chaque carré en quatre, et en affectant des masses m_i suivant les cas. On arrête le processus au niveau du pixel, et on a obtenu une texture extrêmement irrégulière. On peut montrer que la mesure correspondante est fractale et ne dérive pas d'une densité.

Les 4 classes correspondent à quatre choix différents de quadruplets (m_1, m_2, m_3, m_4) et sont présentées figure 7. On a encore 4 photos par classe et 64 échantillons de taille 64×64 par classe. On effectue encore une normalisation statistique (ce qui ne change rien en théorie pour un objet fractal) et on calcule les mêmes paramètres que plus haut. Cette fois, les paramètres sélectionnés sont au nombre de 3, ils sont tous de type fractal. On retrouve ici le fait qu'on a affaire à une bibliothèque fractale (i.e. dont les textures peuvent être analysées à l'aide de paramètres fractals). Ces paramètres permettent une reconnaissance à 100 %. L'arbre est présenté sur la figure 8.

3.3. TEXTURES DE LIÈGE

La troisième bibliothèque se compose d'images plus complexes à classer, puisque cette fois, au lieu de s'intéresser à des textures différentes, on a affaire à diverses déformations d'une même texture de liège. Ces variations dans l'apparence sont obtenues à la suite du processus de fabrication selon les conditions de température et de pression (voir fig. 9).

On voit que même à l'œil, et surtout si l'on ne tient pas compte des différences de niveau de gris moyen, ces textures sont extrêmement difficile à distinguer. Des tests ont montré que le taux de reconnaissance moyen par l'homme est de l'ordre de 70 % sur ces images. Avec les mêmes conditions que celles énoncées en 3.2, nous arrivons au résultat présenté figure 10. Trois paramètres ont été utilisés pour une reconnaissance toujours meilleure que 93 %.

Cette fois, des paramètres « Markoviens » (Haralick) et fractals ont été mélangés, ce qui nous fait classer cette bibliothèque parmi les textures « mixtes ».

On voit sur cet exemple la puissance du système qui nous dit précisément quel type de paramètres il faut faire coopérer, et comment, pour arriver à une classification optimale.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|-----------------------------------|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|----|----|
| Haralick Correlation [0, 4], 2) | 1 | | | | | | | | | | | | | | | | |
| Haralick Correlation [0, 3], 2) | 2 | 99 | | | | | | | | | | | | | | | |
| Autocovariance (5, 0) | 3 | 99 | 98 | | | | | | | | | | | | | | |
| Autocovariance (2, 0) | 4 | 97 | 99 | 96 | | | | | | | | | | | | | |
| Haralick Ctrast (1,1) [0, 3], 2) | 5 | -76 | -76 | -74 | -75 | | | | | | | | | | | | |
| Haralick Ctrast (1,1) [0, 4], 2) | 6 | -75 | -75 | -74 | -74 | 99 | | | | | | | | | | | |
| Haralick IDM (1,1) [0, 2], 2) | 7 | 83 | 83 | 82 | 84 | -94 | -93 | | | | | | | | | | |
| Haralick IDM (1,1) [0, 4], 2) | 8 | 83 | 83 | 84 | 83 | -94 | -93 | 99 | | | | | | | | | |
| Haralick Correlation (0, 1], 2) | 9 | 90 | 92 | 89 | 96 | -66 | -64 | 80 | 77 | | | | | | | | |
| Runlength short run dir 90 dg | 10 | -79 | -79 | -80 | -81 | 78 | 78 | -95 | -94 | -81 | | | | | | | |
| Runlength fract of run dir 90 dg | 11 | -79 | -79 | -80 | -80 | 77 | 76 | -93 | -92 | -81 | 99 | | | | | | |
| Moment centre ordre (0, 2) | 12 | 7 | 8 | 8 | 12 | -50 | -50 | 52 | 51 | 16 | -52 | -50 | | | | | |
| Moment non centre ordre (0, 2) | 13 | 7 | 8 | 8 | 12 | -50 | -50 | 52 | 51 | 16 | -52 | -50 | 99 | | | | |
| Runlength run no-unif dir 45 dg | 14 | -36 | -36 | -36 | -37 | 63 | 63 | -62 | -59 | -36 | 60 | 59 | -27 | -27 | | | |
| Lacunarite L1 [3] s 1% n_poid | 15 | 2 | 1 | 1 | 0 | 41 | 42 | -36 | -37 | 2 | 24 | 22 | -82 | -82 | 1 | | |
| Lacunarite L3 [2] s 2% n_poid | 16 | -4 | -5 | -4 | -5 | 52 | 53 | -43 | -44 | -2 | 30 | 28 | -85 | -85 | 13 | 98 | |
| Lacunarite L1 [1] s 0% n_poid | 17 | -9 | -10 | -9 | -11 | 62 | 62 | -50 | -50 | -7 | 35 | 33 | -83 | -84 | 28 | 91 | 97 |

Figure 5. — Corrélations entre 17 paramètres pour des textures de Brodatz. La demi-matrice inférieure contient les valeurs de corrélation, la demi-

matrice supérieure une représentation de ces valeurs sous forme de niveaux de gris.

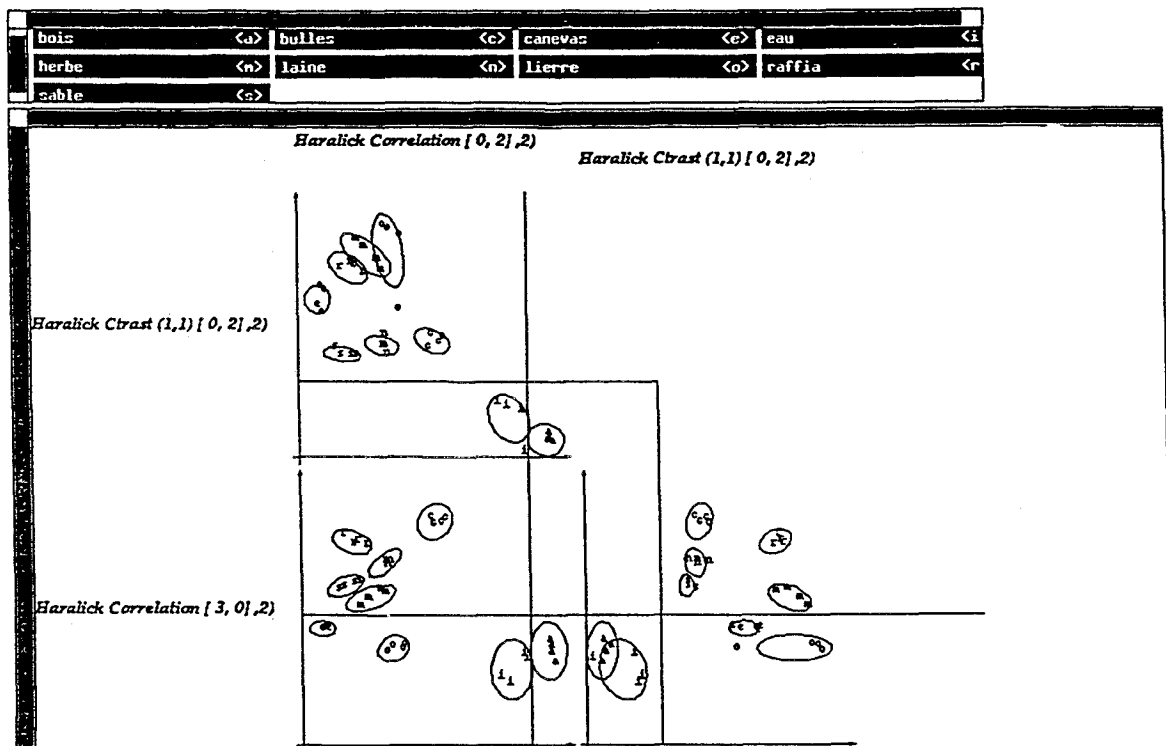


Figure 6. — Multigraphe (représentation par couples de paramètres) à 3 paramètres pour des textures de Brodatz. Chaque lettre représente le barycentre d'une photo. Les ellipses sont les ellipsoïdes d'inertie de

l'ensemble d'apprentissage (une par classe). Les droites sont les valeurs de coupures choisies par DNP.

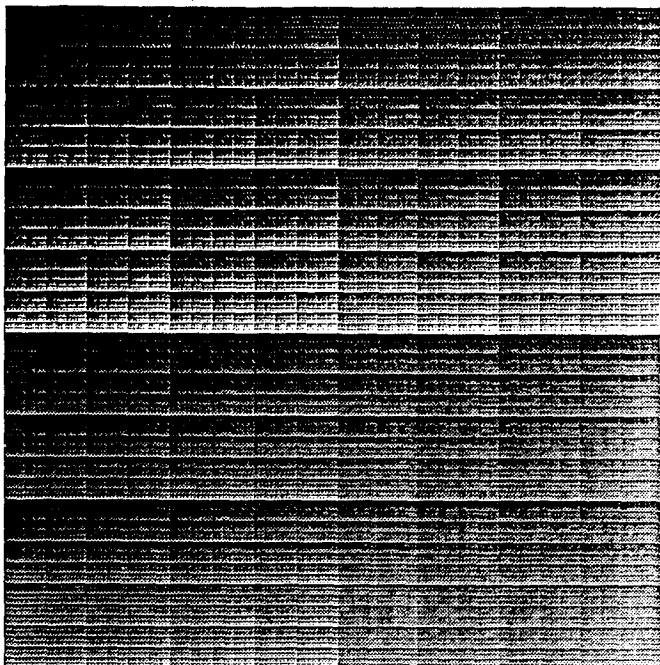


Figure 7. — Textures binomiales.

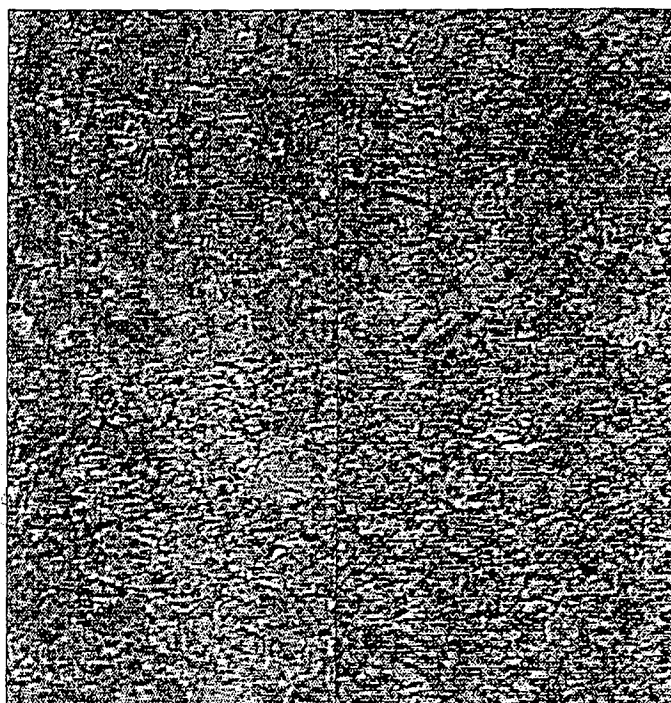


Figure 9. — Textures de liège.

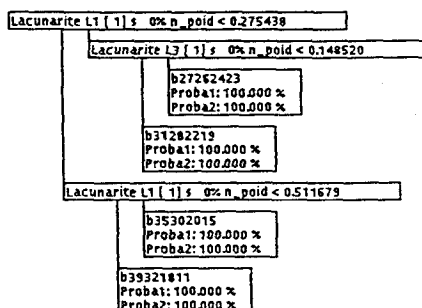


Figure 8. — Arbre de décision pour les textures binomiales.

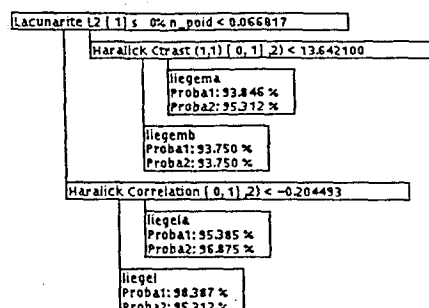


Figure 10. — Arbre de décision pour les textures de liège.

4. Conclusion

Comme on a pu le voir sur les exemples proposés, l'apport d'ARTHUR est d'autoriser les calculs de paramètres divers sur des bibliothèques définies très simplement par l'utilisateur, de façon à :

- classifier les textures et les reconnaître grâce à un arbre de décision ;
- étudier les paramètres présentés dans la littérature, évaluer leur apport spécifique, leur corrélation avec d'autres paramètres, leur pouvoir discriminant ;
- répartir en conséquence les bibliothèques de textures en fonction des classes de paramètres qui les discriminent le mieux et leur sont donc naturellement associés.

ARTHUR a été étendu en un outil de segmentation d'images texturées, EXCALIBUR [21], qui intègre suivant la même philosophie les diverses approches développées à ce jour. EXCALIBUR a permis de segmenter avec une bonne qualité des images complexes (images aériennes, images médicales (voir fig. 11), images radars, etc...) sur la seule base d'informations textuelles, en particulier lorsque les approches de contours sont inutilisables. Une version de ces deux produits sur stations de travail Unix est distribuée par l'INRIA et est actuellement utilisée par plusieurs grandes entreprises et universités.

Manuscrit reçu le 4 février 1992.



Figure 11. — Image RM du thorax et sa segmentation avec EXCALIBUR.

5. Remerciements

Nous remercions particulièrement Jean-Paul Chièze, Nathalie Gaudechoux et Martine Cornélis pour leur aide.

BIBLIOGRAPHIE

- [1] Narendra AHUJA and Azriel ROSENFELD, Mosaic models for textures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(1), 1-11, January 1981.
- [2] Amar AIT-KHEDDACHE and Sarah A. RAJALA, Texture classification based on higher-order fractals. *ICASSP International Conference on Acoustics, Speech and Signal Processing - New York, USA*, pp. 1112-1115, vol. 2, 1988.
- [3] 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.
- [4] L. BREIMAN, J. H. FRIEDMAN, R. A. OHLSEN and C. J. STONE, Classification and regression trees, *Wadsworth*, 1984.
- [5] G. CELEUX *et al.*, Analyse discriminante sur variables continues, *Collection Didactique INRIA*, 1990.
- [6] G. CELEUX, E. DIDAY, Y. GOVAERT, G. LECHEVALLIER and H. RALAMBONDRAINY, Classification automatique des données ; environnement statistique et informatique, *Dunod*, 1989.
- [7] G. CELEUX and Y. LECHEVALLIER, Méthodes de segmentation non paramétriques, *RSA Revue de Statistique Appliquée*, 30(4), 39-53, 1982.
- [8] A. CHHABRA and R. V. JENSEN, Direct determination of the $f(\alpha)$ singularity spectrum. *Physical Review Letter*, 62(12), 1327-1330 March 1989.
- [9] 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.
- [10] J. H. FRIEDMAN, A recursive partitioning decision rule for non parametric classification. *IEEE Trans. on Comp.*, C 26-4, 404-408, 1977.
- [11] André GAGALOWICZ and Christine GRAFFIGNE, Blind texture segmentation. In *Proc. International Conference on Pattern Recognition*. IEEE, October 1988. Rome, Italie, pp. 46-50, vol. 1.
- [12] Mary M. GALLOWAY, Texture analysis using gray level run lengths, *Computer Graphics and Image Processing*, 4, 172-179, 1975.
- [13] Raphaël C. GONZALEZ, Paul WINTZ, Digital image processing *Addison-Wesley* 1977.
- [14] 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 A*, February 1986, 2, 1141-1151.
- [15] Robert M. HARALICK, Statistical and structural approaches to texture, *Proceedings of the IEEE*, 67(5), 786-804, May 1979.
- [16] H. G. E. HENTSCHEL and I. PROCACCIA, The infinite number of generalized dimensions of fractals and strange attractors, *Physica D*, 8(3), 435-444, 1983.
- [17] 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.
- [18] Jacques LÉVY-VEHEL, Analyse et synthèse d'objets bi-dimensionnels par des méthodes stochastiques. *PhD thesis*, Université de Paris-Sud, décembre 1988. ISBN 2-7261-0569-6.
- [19] J. LÉVY-VEHEL, About lacunarity, some links between fractal and integral geometry, and an application to texture segmentation. In *ICCV, Int. Conf. Comp. Vision*, Osaka, Japan, pp. 380-384, 1990.
- [20] J. LÉVY-VEHEL, Fractal probability functions : an application to image analysis. In *CVPR, Comp. Vision and Pattern Recognition*, Hawaii, USA, pp. 378-383, 1991.
- [21] Jacques LÉVY-VEHEL, P. MIGNOT, J. P. BENOIR, Multifractals, Textures and Image Analysis. *CVPR, Comp. Vision and Pattern Recognition*, Champaign, USA, pp. 661-664, 1992.
- [22] Benoît B. MANDELBROT, The Fractal Geometry Of Nature, *W. H. Freeman and company*, 1977.
- [23] B. B. MANDELBROT, Fractal measures (their infinite moment sequences and dimensions) and multiplicative chaos : Early works and open problems. *Technical report*, Physics Department

- ment, IBM Research Center, Mathematics Department. Harvard University, Cambridge, MA 02138, USA.
- [24] B. B. MANDELBROT, A class of multinomial multifractal measures with negative (latent) values for the dimension $f(\alpha)$. In *Fractal (Proceedings of the Erice meeting)*. L. Pietronero, New York, pp. 238-252, 1989.
- [25] C. MENEVEAU and K. R. SREENIVASAN, Measurement of $f(\alpha)$ from scaling of histograms, and application to dynamical systems and fully developed turbulence, *Physics Letter A*, 3, 103-112, March 1989.
- [26] J. MINGERS, An empirical comparison of pruning methods for decision tree induction, *Machine Learning*, 3(4), 319-342, March 1989.
- [27] J. N. MORGAN and J. A. SONQUIST, Problems in the analysis of survey data and a proposal, *J. Amer. Statist. Assoc.*, 58, 415-435, 1963.
- [28] S. PELEG, J. NAOR, R. HARTLEY and D. AVNIR, Multiple resolution texture analysis and classification, *IEEE, PAMI-6*(4), 518-523, July 1984.
- [29] I. PROCACCIA, The characterization of fractal measures as interwoven sets of singularities : Global universality at the transition to chaos, *Springer-Collection : Dimensions and entropies in chaotic systems*, 8-18.
- [30] J. R. QUINLAN, Learning efficient classification procedures and their applications to chess and games, *Machine learning : An artificial intelligence approach*, Morgan kaufman, 1983.
- [31] John R. SULLINS, Distributed learning of texture classification, *ECCV, European Conference on Computer Vision*, Antibes, France, pp. 349-358, 1990.
- [32] Harry VOORHEES and Tomaso POGGIO, Detecting textures and textures boundaries in natural images, *ICCV, Int. Conf. on Comp. Vision*, London, England, pp. 250-258, 1987.
- [33] R. F. VOSS, Random fractal forgeries. In R. A. Earnshaw, editor, *Computer Graphics*, Springer-Verlag Berlin Heidelberg, 1985.
- [34] 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.