# SAMPLE SELECTION IN TEXTURED IMAGES

*Benoît Dolez* [1,2], *Nicole Vincent* [1]

[1] Laboratoire CRIP5-SIP – Université Paris Descartes,
75720 Paris, Cédex 06, France
{benoit.dolez, nicole.vincent}@math-info.univ-paris5.fr

[2] SAGEM Défense Sécurité
178 rue de Paris, 91344 Massy, France
benoit.dolez@sagem.com

## ABSTRACT

This paper proposes a texture learning method based on fractal compression and Iterated Function Systems (IFS). This type of Approach allows to extract self-similarities between blocks of a given image. The number of similarities for each element yields to a score of each blocks. The first blocks of this rating are considered as representative and are stored in a database in order to establish a learning process. Recognition is made by labeling blocks and pixels of the test image. The blocks of the new image are matched with the ones of the different texture databases. As an application, we used our method to recognize bridges and buildings on ground images.

***Index Terms—*** iterated fonction system, fractal compression, learning process, representative blocks extraction, texture

## 1. INTRODUCTION

Image analysis relies on extraction of specific items such as grass, road, wood or bridges. The aim of this paper is to explore a new way for representative area extraction. A particular way is texture analysis, which usually follows one of these approaches: structural, statistical [1], model-based [2] [3], or transform [4]. Low-level features rarely well classify complex concepts. For example, a building contains homogenous *and* geometric areas. Our aim is to take this composite aspect into account by extracting the most representative blocks of the concepts samples. The means we chose is the fractal compression. Fractal theory has been widely worked out during the last two decades ([5], [6], [7], [8], [9]). The basis of this kind of compression is the search for similarities in the image. This type of approach was explored in [10] for handwriting analysis. This paper proposes to continue in this way and extend the previous study to grey scale images and texture learning.

## 2. REPRESENTATIVE BLOCKS MATCHING

### 2.1 Our method principle

Our aim is to extract representative elements in the image in order to build a learning process. We consider redundant elements in a signal as representative and so we want to establish associations between blocks in order to measure redundancies in the image. These associations are to be independant from certain criteria such as rotation, symetry, scale, contraste and luminosity. Such properties can be found in Iterated Function Systems.

### 2.2 Iterated fonction systems principle

Iterated Function Systems (IFS) are a basis of fractal image compression. An IFS is a set of contracting functions $T_i : M \to M$ in a metric space $M$. This contracting function can be extended to the set of parts of $M$ and considering the Hausdorff distance.

$$T = \bigcup_{i=1}^{N} T_i : P(M) \to P(M)$$

The fix point theorem gives the existence and uniqueness of a subset $F$ of $M$ so that $T(F) = F$. $F$ is called the attractor of the IFS. An image is rarely self-similar, for this reason, we use the principle of Partitioned IFS (PIFS): the image is partitioned, then for each element of the partition we aim at finding an area of the image which should correspond, apart from a contracting transform. In our case the partition is a regular grid. Its elements are called *ranges*. We have chosen them square. The zones of the image that may correspond are called *domains*.

The decompression step of the image is limited to the application of the PIFS. The initial point is an image that has the same dimension as the compressed image but any content is convenient (average grey image for example or an ordinary image). At each iteration, the image is transformed and converges towards the image compressed by the PIFS. Many papers have been published in this field during the last two decades. Some of them deal more specially with the memory size required to encode the compressed image [11]. Others study the optimization of correspondence search between similar elements in the image ([9] , [12], [13], [14], [15], [16]).

### 2.2 Iterated fonction system construction

To ensure contracting property, the magnitude of the scale factor between a domain and a range, and the contrast parameter must be lesser than one. Finding the contracting transform is equivalent to find, for each couple of range $R$ and domain $D$, the parameters that best match the following expression:

$$R \cong iso(res(D,s), iIso).c + l$$

where $red(D,s)$ is the result of scaling $D$ by a $s$ factor, $iso(B,iIso)$ is the application of an isometry referred by $iIso$ on image block $B$. $c$ and $l$ are contrast and luminance parameters and can be computed by least square method. To respect the square shape of the blocks we limit the considered isometries to the eight possibilities of rotations and symmetries ($iIso \in [1..8]$). In practice, the scale factor can be fixed to $1/2$ as explained by Jacquin in [6]. We say that domain $D$ encodes range $R$ if the matching between $D$ and $R$, by $T = (D, R, iIso, s, c, l)$, is good enough, that is to say they can be considered as similar according to some given criterion: PSNR, RMS, etc.

## 3. SELECTION PROCESS

### 3.1 Domain score definition and optimization

The compression phase allows knowing which part of the image encodes which other one. To define the score of a domain, the main idea is to know how many ranges may encode the domain, according to some reconstruction error threshold $S$. The reconstruction error of a range $R$ by a domain $D$ according to a given transform $T$ is defined by

$$Er(D,R) = \min_{s,iIso,c,l}\left[d\left(iso(res(D,s), iIso).c + l, R\right)\right]$$

Where $d(.,.)$ is the RMS measure between two blocks.

Then, we have $score(D,S) = \sum_R \delta_{Er(D,R)\leq S}$ where

$$\delta_P = \begin{cases} 1 \; if \;\; P \; is \;\; true \\ 0 \; otherwise \end{cases}$$

So we define a score to rate domains. The score of $D$ is the number of ranges $R$ that verify $Er(D,R) \leq S$. We may notice that when contrast is low, the matching between blocks cannot be considered as representative. As natural images are coherent, the domain score map associated with the image varies practically in a continuous way. Then we can say the pertinence of a domain is not located exclusively on this domain location but expands to its neighborhood. This phenomenon ensures the robustness of our method, regardless of small variations according to the choice of a partition of the initial image.

An exhaustive run of the range/domain couples can be computionaly expensive for large image area. A possible optimization, or restriction, is to rely on a measure and for a given range (resp. domain), only take into account domain (resp. ranges) that have a similar measure. Such restriction measure could be fractal dimension or entropy like proposed in [17].

We have just seen how to rate domains. This allows us to establish an order relation among the domains and gives us a choice criterion, so that the learning process leads to a good modeling of the image or texture.

### 3.2 Domain sorting and representativness definition

We want to characterize textures through most representative domains. Selecting only domains with the highest score is not enough as, from the previous remark, two neighboring domains may have similar scores and code parts of the same zone of the image. In fact, we want to encode the *biggest possible area of the image with the smallest number of domains*. In order to solve this problem, we establish an increasing threshold procedure. Then domains are selected as follows:

1. All ranges are marked *not encoded*
2. We compute the score of each domain according to actual $S$, taking into account only non-encoded ranges
3. The domain with the best score is selected and stored. All the ranges it encoded are marked *encoded.*
4. Repeat 2 and 3 until the needed percentage is reached.

In the practical case, matrix $Er(.,.)$ must be rounded, clusterized, in order to have enough couples in competion at each step. The smaller the clusters are, the finest the reconstruction will be.

### 3.3 Inter class discrimination

In order to achieve the segmentation of an image, several textures have to be learnt using the previous method. When learning a texture, some domains can be ambiguous as they allow encoding parts of other textures. These domains may be prejudicial to the discrimination. To suppress this ambiguity, we try to reconstruct a texture (i.e. its ranges) with domains coming from other textures. A domain from a texture is considered as ambiguous if it allows encoding more than a given percentage of ranges of another texture. The ambiguous domains are suppressed from the learning data base.

## 4. EXPERIMENTATION

### 4.1 Diversity and similarity measure

For each learnt texture, we have a set of representative domains we will try to find in the test image. A redundent problem when using block matching approaches is the importance of taking neighbourhood into account in order to suppress false alarms. To include this constraint we suppose that one can be more confident in an area reconstructed with *many different* domains of a concept rather than another area reconstructed with a very few of them. This parameter is called *richness* of domains. Let $c_i$ be a learned concept and $p$ a pixel on the test image, $rich(p, c_i)$ is the number of different domains of $c_i$ used to reconstruct the neighborhood of $p$. The more the richness is high and the distance is small, the more we are confident in the labeling of the considered area.

### 4.2 Distance definition for the recognition task

The principal originality of this study is the learning process. Once the reference block database has been computed for each concept, the local recognition task can be summarized as a classical search of the nearest element between the blocks of the test image and each domain of each concept. The local distance is computed by comparing normalized block attributes. When using both richness and local distance parameter, we must normalize them in order to compute a coherent value. Let $p$ be a particular pixel. For each class $c_i$ we know the normalized distance $\overline{dist}(p, c_i)$ and the normalized richness $\overline{rich}(p, c_i)$. The final distance of $p$ is defined as

$$d_{final}(p, c_i) = \sqrt{\overline{dist}(p, c_i)^2 + \left(1 - \overline{rich}(p, c_i)\right)^2}$$

And so the label of $p$ will be:

$$label(p) = \arg\min_i \left(d_{final}(p, c_i)\right)$$

### 4.3 Learning data and test results

We have chosen to learn two classes: building and vegetation. In the learning image, areas 1 and 2 correspond to vegetation, and areas 3 and 4 correspond to buildings.

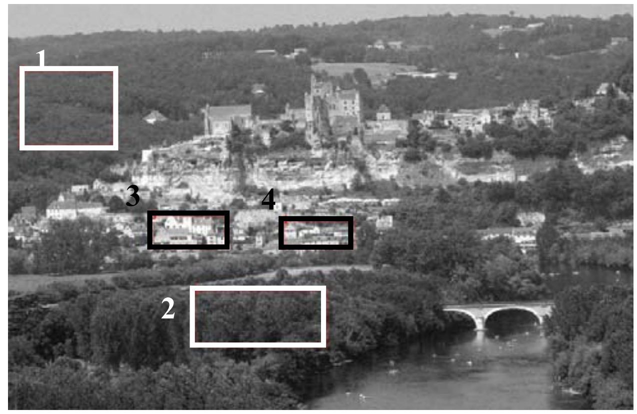


**Fig. 1.** Learning data base sample. Areas 1 & 2 contain vegetation; areas 3 & 4 contain buildings (private source image).

**Fig. 2.** Test image (public source image).

**Fig. 3.** Ground truth of the test image: white for buildings, black for vegetation.

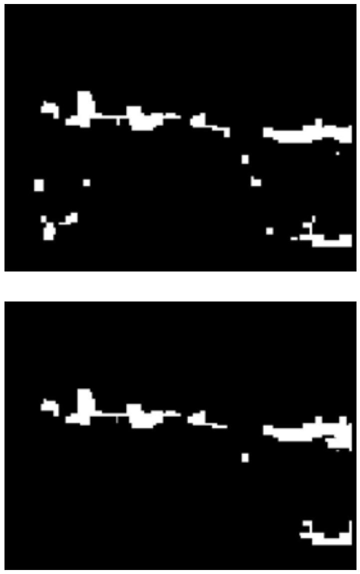

**Fig. 4.** Buildings recognition without (upper) and with (lower) richness parameter on the left and right respectively.

Richness parameter ollow to suppress some false alarms, notably on the bottom left and right. Some alarms on the bottom right remained. In this configuration, the building recognition match better with the ground truth. We notice that the small isolated building was not eliminated and is still recognized.

## 5. CONCLUSION

We have seen how to use the principle of fractal compression to build a learning process for texture concept by extracting similar blocks. Then we presented a recognition step and finally some test results of our method. We highlighted the importance of taking the neighborhood of each pixel into account. In particular, this notably allowed to remove false alarms and to get our recognition nearest to the ground truth thanks to the domain richness parameter. One may notice that parameters such as the number of reference blocks of each concept can be set to match the system's constraints (especially memory size, computing unit speed), for embedded systems and robotic use for example.

## 6. REFERENCES


[1] R. Haralick, "Statistical and Structural Approaches to Texture", *Proc. IEEE*, Vol. 67, n°5, pp. 786-804, 1979.

[2] A. Pentland, "Fractal-Based Description of Natural Scenes", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 6, n°6, pp.661-674, 1984.

[3] L. Kaplan and C-C Kuo, "Texture Roughness Analysis and Synthesis via Extended Self-similar (ESS) Model", *IEEE Trans Pattern Analysis and Machine Intelligence*, Vol. 17, n°11, pp. 1043-1056, 1995.

[4] L. Lepisto, I. Kunttu, J. Autio, A. Visa, "Classification method for colored natural textures using Gabor filtering", *Image Analysis and Processing*, Proceedings 2003, pp. 397 – 401.

[5] M. Ghazel, G. H. Freeman, and E. R. Vrscay, "Fractal Image Denoising", *IEEE Transactions on Image Processing*, Vol. 12, N° 12, décembre 2003.

[6] A. Jacquin, "Fractal Image Coding Based on a Theory of Iterated Contractive Image Transformations", *Proc. SPIE Visual Comm. And Image Proc.*, pp. 227-239, 1990.

[7] A. Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations", *IEEE Transactions on Image Processing*, Vol. 1, pp. 18-30, 1992.

[8] Y. Fisher, *Fractal Image Compression – Theory and Application*. New York : Springer-Verlag, 1994.

[9] M. R. J. Maria, S. Benoît, B. Macq: "Speeding up fractal image coding by combined DCT and Kohonen neural net method", *ICASSP'98 - IEEE Intl Conference on Acoustics, Speech and Signal Processing*, Proc., pp. 1085-1088, May 1998.

[10] N. Vincent, A. Seropian, and G. Stamon, "Synthesis for handwriting analysis", *Pattern Recognition Letters n°26*, pp. 267-275, December 2004.

[11] H. T. Chang, "Gradient Match and Side Match Fractal Vector Quantizers for Images", *IEEE Transactions on image processing*, Vol. 11, n°1, January 2002.

[12] Lai, Lam, Siu, "A fast image coding based on kick-out and zero contrast conditions", *IEEE Transactions on image processing*, Vol. 12, n°11, November 2003.

[13] R. Hamzaoui, "Codebook clustering by self-organizing maps for fractal image compression", *Institut für Informatik – Report 75*, January 1996.

[14] B.E. Wohlberg and G de Jager, "Fast Image Domain Fractal Compression by DCT Domain Block Matching", *Electronics Letters*, Vol. 31, pp. 869-870, May 1995.

[15] R. Distasi, M. Nappi and D. Riccio, "A Range / Domain Approximation Error-Based Approach for Fractal Image Compression", *IEEE Transactions on Image Processing*, Vol. 15, N° 1, January 2006.

[16] K.U. Barthel, J. Schüttemeyer, T. Voyé and P. Noll, "A New Image Coding Technique Unifying Fractal and Transform Coding", *IEEE International Conference on Image Processing*, Novembre 1994.

[17] M. Hassaballah, M.M. Makky, Y.B. Mahdy, "A Frast Fractal Image Compression Method Based Entropy", *Electronic Letters on Computer Vision and Image Analysis*, February 2005.