FRACTAL COMPRESSION OF IMAGES WITH PROJECTED IFS

Éric Guérin, Éric Tosan and Atilla Baskurt

LIRIS, Université Claude Bernard Lyon 1 – 8, bd Niels Bohr, 69629 Villeurbanne Cedex
eguerin@ligim.univ-lyon1.fr

ABSTRACT

Standard fractal image compression, proposed by Jacquin[1], is based on IFS (Iterated Function Systems) defined in $\mathbb{R}^2$. This modelization implies restrictions in the set of images being able to be compressed. These images have to be self similar in $\mathbb{R}^2$. We propose a new model, the projected IFS, to approximate and code grey level images. This model has the ability to define affine IFS in a high dimension space, and to project it through control points, resulting in a non strictly self similar object in $\mathbb{R}^2$. We proposed a method for approximating curves with such a model [2, 3]. In this paper, we extend the model capabilities to surfaces and images. This includes the combination of projected IFS in a quadtree structure and a complete coding scheme. First results show that our method gives better results than standard fractal image compression. Furthermore, in the very low bitrate context, the distortion/rate performances are equivalent to those obtained with EZW algorithm.

1. INTRODUCTION

Performing even better compression on images has always been an important research area. From the old standard JPEG to JPEG 2000, many improvements have been supplied. In 1992, Jacquin proposes a fractal compression algorithm that uses the spatial redondancy of natural images using Iterated Function Systems (IFS) [1]. In this study we introduce a new model for fractal compression of images using combination of projected IFS in a quadtree structure. We first explain the projected IFS model and then give extensions with quadtrees and the compression method. We finally expose our first compression results and compare it with other compression methods (JPEG, JPEG 2000 and Jacquin compression method).

2. PROJECTED IFS APPROXIMATION

2.1. IFS

Introduced by BARNSLEY[4] in 1988, the IFS (Iterated Function Systems) model generates a geometrical shape or an image [1] with an iterative process. An *IFS* $\mathbb{T}$ *(Iterative Function System)* is a finite subset of contractions in a metric space $(\mathcal{X}, d)$: $\mathbb{T} = \{T_0, ..., T_{N-1}\}$.

We note $\mathcal{H}(\mathcal{X})$ the set of non-empty compacts of $\mathcal{X}$. The associated HUTCHINSON operator is:

$$K \in \mathcal{H}(\mathcal{X}) \quad \mapsto \quad \mathbb{T}K = T_0 K \cup ... \cup T_{N-1} K$$

This operator is contractive in the new complete metric space $\mathcal{H}(\mathcal{X})$ and admits a fixed point, called *attractor* [4]:

$$\mathcal{A}(\mathbb{T}) = \lim_{n\to\infty} \mathbb{T}^n K \text{ with } K \in \mathcal{H}(\mathcal{X})$$

By introducing an index set [4], and adding joining conditions [5, 6, 7], the attractor defines parametric surfaces. For surfaces or images, it is convenient to use a double indexing $\Sigma = \{0, \ldots, N-1\} \times \{0, \ldots, N-1\}$ [8]:

$$\Phi(s,t) = \phi(\rho) \text{ with } \rho = (\sigma_1, \tau_1) \ldots (\sigma_n, \tau_n) \ldots \in \Sigma^\omega$$

where $\sigma = \sigma_1 \ldots \sigma_n \ldots$ and $\tau = \tau_1 \ldots \tau_n \ldots$ are respectively the development of s and t in base N.

2.2. Projected attractors

The main idea of our model is drawn from the formula of free form surfaces used in CAGD (Computer Aided Geometric Design):

$$F(s,t) = \sum_{j\in J} \Phi_j(s,t) p_j$$

where p_j constitutes a grid of control points, and Φ_j are blending functions. These blending functions have the following property:

$$\forall (s,t) \in [0,1]^2 \quad \sum_{j\in J} \Phi_j(s,t) = 1$$

In classical fractal interpolation [4, 9] or fractal compression [1], the complete metric space $\mathcal{X}$ used is $\mathbb{R}^2$ or $\mathbb{R}^3$, and the iteration semigroup is constituted of contractive affine operators. Our work consists in enlarging iteration spaces [5, 6]. This model uses a barycentric space $\mathcal{X} = \mathcal{B}^J$:

$$\mathcal{B}^J = \{(\lambda_j)_{j\in J} \mid \sum_{j\in J} \lambda_j = 1\}$$

For surfaces or images, we use $J = \{0, \ldots, m\} \times \{0, \ldots, m\}$. Then, the contractions are matrices with barycentric columns:

$$S_J = \{T \mid \sum_{j \in J} T_{ij} = 1, \forall i \in J\}$$

This choice leads to the generalization of IFS attractors named *projected IFS attractors* (PIFS):

$$P\mathcal{A}(\mathbb{T}) = \{P\lambda \mid \lambda \in \mathcal{A}(\mathbb{T})\}$$

where P is a grid of control points $P = (p_j)_{j \in J}$ and $P\lambda = \sum_{j \in J} \lambda_j p_j$. In this way, we can construct a fractal function [5, 6, 7] using the projection:

$$F(s,t) = P\Phi(s,t) = \sum_{j \in J} \Phi_j(s,t) p_j$$

where $\Phi(s,t)$ is a vector of functions $\Phi(s,t) = (\Phi_j(s,t))_{j \in J}$ and J is the double index set $J = \{0, \ldots, m\} \times \{0, \ldots, m\}$.

Fig. 1 shows the difference between action on the control grid P which is global deformation and action on $\mathbb{T}$ which involves local regularity of the surface. Note that a surface representation has been used for better understanding.

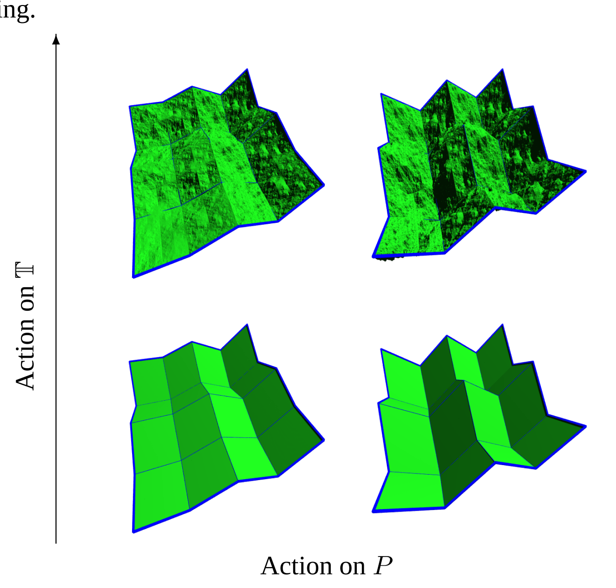


Fig. 1. Global deformation using the control grid P and local aspect transformation using $\mathbb{T}$

2.3. Approximation with projected IFS (PIFS)

Considering the coefficients of the subdivision matrices and the coordinates of the control points are put all together in a parameter vector $\mathbf{a}$, it is possible to construct a fractal curve (or surface) family:

$$F_{\mathbf{a}}(s,t) = P_{\mathbf{a}} \Phi_{\mathbf{a}}(s,t)$$

From now on, the approximation problem is equivalent to the following minimization problem:

$$\mathbf{a}_{opt}(\mathbf{Q}) = \operatorname*{argmin}_{\mathbf{a}} \mathcal{E}(F_{\mathbf{a}}, \mathbf{Q})$$

where $\mathbf{Q}$ is the original data to approximate and $\mathcal{E}$ is a function that gives the numerical spacing between two surfaces or images.

For image approximation, we use scalar control points $p_j \in \mathbb{R}$, and a simple L^2 distance as spacing function. Furthermore, computation is more efficient with such simplifications because it is possible to describe the model with a simplified subdivision scheme [10]. Numerical optimisation is performed by a standard Levenberg-Marquardt algorithm. For detailed results, see [10].

Results show that this single model approximation is good but rather limited to small image data. For natural size images (say for example 512×512), it would require extremely complex model involving a great number of parameters that could not be retrieved by a minimization algorithm. Moreover, it would be illusory to modelize a whole image with a unique model. A better method should be to divide the problem into smaller ones we are able to solve. This is the goal of our new model that combines PIFS in a quadtree structure in order to better fit to local characteristics.

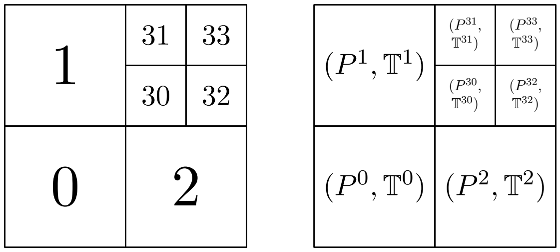
3. COMBINATION OF PIFS

In this section we introduce a quadtree structure for combining models and providing heterogeneous modelization capabilities.

3.1. Quadtree structure

Let us denote Γ the cut of a quadtree and $\gamma \in \Gamma$ a leaf of this cut. γ is a finite word of $\{0,1,2,3\}$. Fig. 2a gives a quadtree cut example, defined by:

$$\Gamma = \{0, 1, 2, 30, 31, 32, 33\}$$



(a) Quadtree cut example

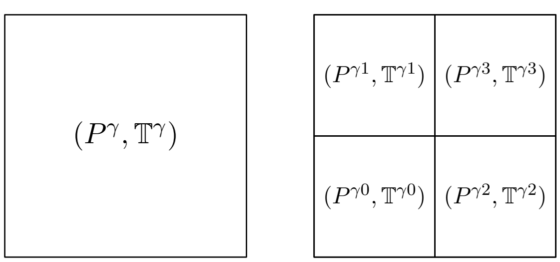
(b) Projected IFS quadtree example

Fig. 2. Quadtree examples

Now imagine that we associate to each leaf γ a complete PIFS model: $(P^\gamma, \mathbb{T}^\gamma)$. We obtain a quadtree of PIFS illustrated in Fig. 2b.
If we want to obtain a continuous surface or image, we have to add joining conditions between each leaf model. Because these constraints involve complex relations, it is really difficult to take them into account. That's the reason why we have chosen to leave each leaf model completely independant. In an image context, it implies an arbitrary choice of which leaf a border pixel comes from.

3.2. Refinement

This new hierarchical model implies more adaptability to the data entries. Imagine a leaf γ which is approximating a part of an image. If this approximation is not sharp enough, we have the ability to *refine* it (subdivision). That means replacing it with four subdivided leaves $\gamma0$, $\gamma1$, $\gamma2$ and $\gamma3$. Fig. 3 shows the initial model with a single leaf (a) and the refined model with four leaves (b).



(a) Initial model

(b) Refined model

Fig. 3. Refinement principle

3.3. Exhaustive exploration

Given an image to approximate, the method developed for constructing a PIFS quadtree has four steps: exhaustive exploration of the approximation quadtree, adaptive quantification of the different models, coding and optimization.

The first step uses a recursive algorithm that first takes the whole image and approximates it with a single PIFS. Then, image is subdivided in four parts, and approximation is performed on each part. And so on, recursively, with images becoming even smaller. At a particular depth, we stop the algorithm to avoid approximation of a to small image. Intermediate approximation results are stored in an adapted structure: a quadtree. For a better adaptation, different kinds of PIFS models for each image are considered. Four models are used ranging from 3×3 control points with 36 parameters to 9×9 control points with 192 parameters.

3.4. Quantification and coding

For more efficiency, we have chosen to perform an adaptative quantification. This implies we have to modelize statistic distribution of the model parameters, which is in fact essential for coding too. Two kinds of parameters must be considered for statistic modelization: parameters representing control points and the subdivision coefficients.

For the first category, the distribution is completely correlated with the grey levels distribution of the image we analyze. Distribution of the control points is modelized by the sum of four generalized gaussians as shown in Fig. 4.

Concerning the distribution of the subdivision coefficients, it is less dependant of the image we are dealing with. Fig. 5 shows the approximation of the distribution with a single generalized gaussian.

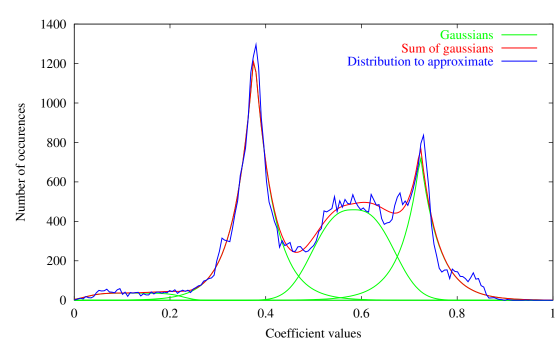


Fig. 4. Approximation of the control points distribution with four generalized gaussians

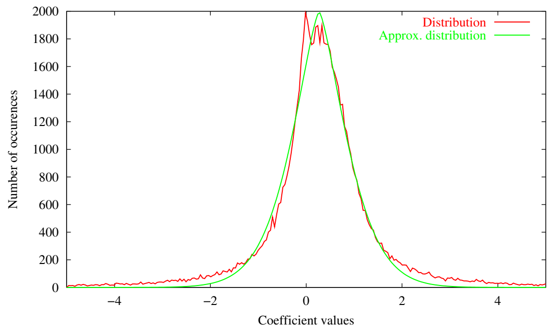


Fig. 5. Approximation of the subdivision coefficients distribution with a generalised gaussian

Quantification is performed with different number of quantification intervals, ranging from 2^2 to 2^9. After each quantification, arithmetic coding of the two kinds of parameters is performed with the modelized distributions.

3.5. Distortion/rate optimization

We now have a large number of possible descriptions for the image. Each description involves a choice of a quadtree cut, a model for each leaf, and two quantifications per model: one for the control points, one for the subdivision coefficients. Each of these descriptions implies a rate (computed by adding the code lengths) and a distortion with respect to the original image. Due to the data structure, it is possible to get rid of the large multiplicity of descriptions by using a lagrangian multiplier paradigm. Indeed, the minimisation of the lagrangian cost can be performed using a recursive run through the quadtree. We can now use standard optimisations methods for finding the lagrange multiplier that satisfies a constraint. The algorithm used is drawn from [11].

3.6. Results

In this section, we present the very first results obtained with PIFS.

Fig. 7 shows the original image, a portion of the well-known *peppers* picture. Fig. 6 gives Rate/Distortion curves for this image with four compression methods: Jpeg, fractal compression, EZW and PIFS. The proposed method is interesting for very low bit rates (less than 0.1 bpp). For these rates, the distortion results are equivalent to those of EZW. Note that our method outperforms the classical fractal compression. Moreover, PIFS is better than JPEG for bitrates lower than 0.2 bpp. This result has been observed with all the other images tested (*lena*, *airplane*, *baboon*, *gold*, *boats*…). Fig. 8 and 9 show this image compressed at 0.044 bpp with two methods: EZW and our method. For this high compression ratio (180), the visual quality of the method PIFS clearly appears. At this rate, EZW generates artefacts along contours. PIFS provides well-defined contours with less visual ringing effects. Note that a simple post-processing step, 3×3 gaussian blur filter, improves the visual and numerical result (see Fig. 10).

Our method allows to reconstruct perfectly the original image by going deeper in the quadtree exploration but it's not competitive compared to other methods.

Average computing time for 256×256 images is 15 minutes, and about one hour for a 512×512 image (Athlon 1.2 GHz hardware). These computing times include compression for a large number of bitrates.

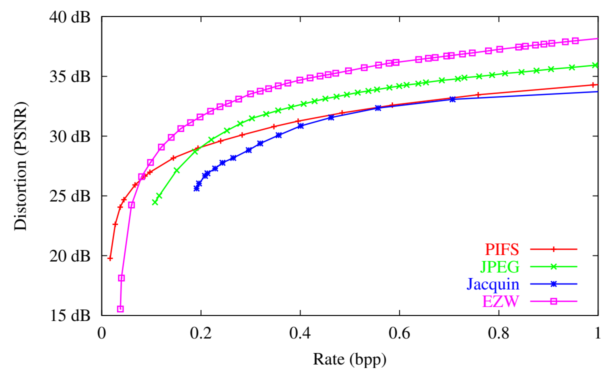


Fig. 6. Rate/Distortion curves

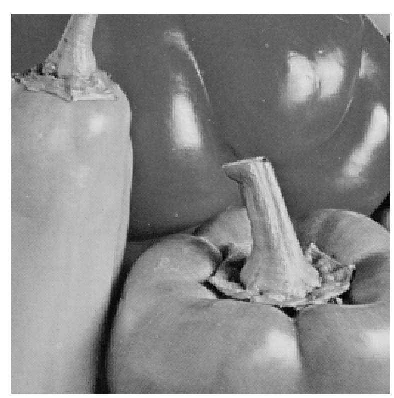


Fig. 7. Original image, portion of *peppers*

Fig. 8. Image compressed at 0.044 bpp with EZW, PSNR $= 23.1dB$

Fig. 9. Image compressed at 0.044 bpp with PIFS, PSNR $= 24.7dB$

Fig. 10. Image from Fig. 9 processed with a 3×3 gaussian filter, PSNR $= 25.1dB$

4. CONCLUSION

A new fractal compression scheme is introduced using a completely different model from the former *fractal image compression*. Instead of using IFS's with transformation in $\mathbb{R}^2$, we define them in a higher dimension space and then project them: "projected IFS model". Combining a set of PIFS in a quadtree structure, it is possible to model complex images with spatially varying properties. With a complete optimal coding scheme, we have tested our model for compression purpose. It is more efficient than fractal image compression and is equivalent to EZW in very low bitrates. The visual quality obtained with PIFS is slightly better.

5. REFERENCES

[1] A E Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Trans. on Image Processing*, 1:18–30, January 1992.

[2] Eric Guérin, Eric Tosan, and Atilla Baskurt. Fractal coding of shapes based on a projected IFS model. In *ICIP 2000*, volume II, pages 203–206, September 2000.

[3] Eric Guérin, Eric Tosan, and Atilla Baskurt. A fractal approximation of curves. *Fractals*, 9(1):95–103, March 2001.

[4] Michael Barnsley. *Fractals everywhere*. Academic Press, 1988.

[5] Chems Eddine Zair and Eric Tosan. Fractal modeling using free form techniques. *Computer Graphics Forum*, 15(3):269–278, August 1996. EUROGRAPHICS'96 Conference issue.

[6] Chems Eddine Zair and Eric Tosan. Computer Aided Geometric Design with IFS techniques. In M M Novak and T G Dewey, editors, *Fractals Frontiers*, pages 443–452. World Scientific Publishing, April 1997.

[7] Eric Tosan. Surfaces fractales définies par leurs bords. In L. Briard, N. Szafran, and B.Lacolle, editors, *Journées "Courbes, surfaces et algorithmes", Grenoble*, September 1999.

[8] C A Micchelli and H Prautzsch. Computing surfaces invariant under subdivision. *Computer Aided Geometric Design*, (4):321–328, 1987.

[9] P. Massopust. *Fractal Functions, Fractal Surfaces and Wavelets*. Academic Press, 1994.

[10] Eric Guérin, Eric Tosan, and Atilla Baskurt. Modeling and approximation of fractal surfaces with projected IFS attractors. In M. M. Novak, editor, *Emergent Nature*. World Scientific, 2002.

[11] Kannan Ramchandran and Martin Vetterli. Best wavelet packet bases in rate-distortion sense. *IEEE Transactions on Image Processing*, 2(2), April 1993.