Fractals, Vol. 5, Supplementary Issue (April 1997) 111–123

ZOOMING USING ITERATED FUNCTION SYSTEMS

ERIC POLIDORI and JEAN-LUC DUGELAY
Institut EURECOM, Multimedia Communications dept.
2229, route des Crêtes, B.P. 193
06904 Sophia Antipolis, FRANCE

Abstract

Iterated Function Systems (I.F.S.) have been recently studied in the field of image coding. In addition to compression, I.F.S. possess some properties of fractals which can be used to sub/oversample an image. In this paper, we review how I.F.S. can be used for image zooming, directly from basic algorithms of still image compression. This study shows that results obtained in such a way do not produce better results than those obtained by using classical spatial interpolators such as the linear one. However, in this paper we also show that modified versions of the basic approach provide a better use of I.F.S.'s coding for zooming.

1. INTRODUCTION

Since 1990, I.F.S. have been studied in the field of image coding. The reference algorithm, from Arnaud Jacquin, is briefly reviewed in section 3. Since then, several relevant papers proposed some improvements. Moreover, some authors indicate that the fractal code built during the coding stage is independent of the size of the original image, and then this code can be used to reconstruct an image at any level of resolution without a major loss of definition. This mechanism is detailed in section 4.

Currently, in several fields, oversampling is often needed. In aerial or satellite imaging, zoom is used in order to facilitate the image interpretation, or just to obtain a more comfortable visualization environment. In some multimedia applications, such as image databases consultation, sub/oversampling tools are often required in order to deliver a reduced (i.e. icon) or an enlarged (i.e. zoom) version of a given image. Available software realizes zooms using some classical interpolators, such as nearest-neighbor (duplication), linear or cubic interpolation. These interpolators are briefly presented in section 2.

In the last section, a comparison shows that images obtained by the classical fractal zoom do not give better results than those obtained by using classical interpolators. Nevertheless, improvements and modifications of the basic algorithm using I.F.S., proposed in section 5, allow to define an effective alternative to classical interpolators for image zooming.

Illustrations on the well-known image "Lenna" are presented in section 6, as well as a diagram (Fig. 13) which shows, in a subjective way for the visual criterion and rigorously for the others, the global classification of some methods.

2. CLASSICAL INTERPOLATORS

The following interpolators are particular oversamplers. An interpolator (or interpolation function) is a function which is equal to another function for some points (interpolation nodes). That is the main difference between the fractal oversamplers, described in the following sections, which do not necessarily keep the original luminance values. Each interpolator used in this work is a polynomial function.

The simplest oversampling is the Nearest-Neighbor Interpolation (N.N.I.). It consists in duplicating the original pixels' values. For example, when zooming by a factor two, each original pixel is duplicated four times. So, the degree of the polynomial function of interpolation is zero.

In practice, the most frequently used oversampling is the Linear Interpolation (L.I.). This interpolator is based on a local hypothesis of luminance signal continuity and calculates, by averaging[1], a value at a subpixel position.

The last interpolator we used as a reference is a modified version of the cubic one, the Cubic Convolution Interpolation[2] (C.C.I.).

Nearest-neighbor, linear and cubic convolution interpolators, which are approximations of the first, second and third orders respectively, are based on local continuity hypothesis of the luminance signal and use a set of pixels located in a neighborhood (of resp. 1, 4 and 16 pixels) around the position to be interpolated.

3. FRACTAL CODING STAGE (GENERALITIES)

Fractal coding of a still image μ_{orig} consists in building a code τ (i.e. a particular transformation) such that μ_{orig} is **approximately self-transforming** under τ (i.e. $\mu_{orig} \approx \tau(\mu_{orig})$). Then, if τ is a contractive transformation, μ_{orig} is approximately the attractor of τ (i.e. $\mu_{orig} \approx \tau^{\infty}(\mu_0)$ for some initial image μ_0, if we write $\tau^{\infty} = \lim_{k\to\infty}\tau^{ok}$). This code τ is built on a partition of the original image. Each block R_i of this partition is called a **range block** and is coded independently of the others by a matching (local code τ_i) with another block D_i in the image, called a **domain block**. According to the Jacquin's coding algorithm[3], let us note $\tau(\mu_{orig}; R, n, p_x, p_y, \alpha_m)$ the fractal code obtained by using the parameters :

- R the range blocks' size (in case of squared range blocks)

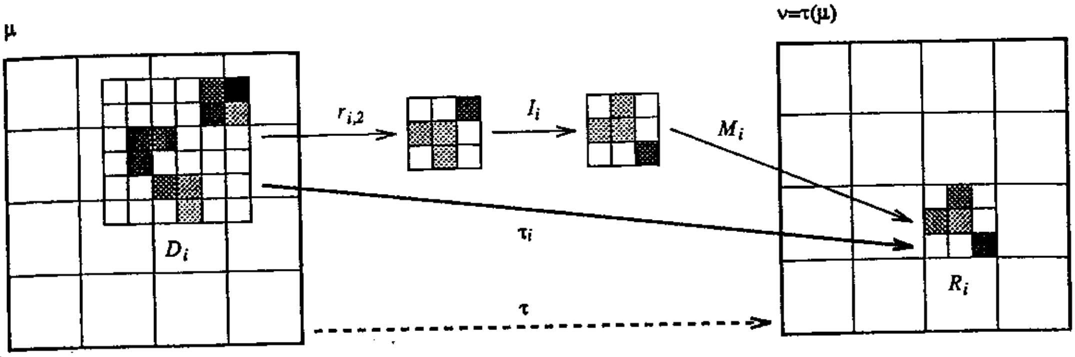


Fig. 1 τ_i **behavior.**

- n the scale factor used for the local self-similarity search. If D is the domain block's size, then $D = n \cdot R$.
- p_x (respectively p_y) the horizontal (resp. vertical) step which defines the domain pool (set of all the domain blocks) for an exhaustive research. The coordinates of each domain block's upper left corner are expressed by $(j \cdot p_x, i \cdot p_y)$, where i and j are integers.
- α_m the upper boundary on the scales (of the massic transforms). More precisely, $\tau = \bigcup_{i=1}^{N} \tau_i$ where $\tau_i : D_i \longrightarrow R_i$ and $\tau_i = M_i \circ I_i \circ r_{i,n}$ with $M_i(x) = a_i \cdot x + b_i$ an affine operator with a scale a_i and a shift b_i on the luminance of the pixels, I_i a transformation selected from eight discrete isometries and $r_{i,n}$ a reduction by a factor n using an averaging (see Fig. 1). So, during the coding stage, $0 \leq a_i < \alpha_m, \forall i \in \{1, ..., N\}$.

In this study, $n = 2$, so $D = 2R$ and $r_{i,2}$ is a reduction by a factor two.

4. FRACTAL ZOOM (F.Z.)

4.1 Classical Fractal Decoding Stage

The decoding stage, based on a continuous theory (I.F.S.), consists in an iterated process. It needs an arbitrary initial image μ_0 and the fractal code τ (see Fig. 2). Throughout this paper, the initial image μ_0 has black pixels. Then, the τ's attractor $\tau^\infty(\mu_0)$ gives an approximation of the original image μ_{orig}. By denoting ε_c the coding error, the reconstruction error ε_r is bounded, according to the **Collage theorem**, by the following expression :

$$\varepsilon_r \leq \frac{\varepsilon_c}{1-s} \tag{1}$$

where $\varepsilon_c = d_2(\mu_{orig}, \tau(\mu_{orig}))$, $\varepsilon_r = d_2(\mu_{orig}, \tau^\infty(\mu_0))$ and s the contractivity of the transformation τ. In this context, the contractivity constraint is expressed as : for all images μ and ν, $d_2(\tau(\mu), \tau(\nu)) \leq s \cdot d_2(\mu, \nu)$ with $s \in [0, 1[$ (d_2 is the Euclidian metric).

4.2 Decoding With Zoom

The main idea of the fractal zoom is rather simple. It is based on an important property of the fractals. If we assume that fractal coding is really a fractal process, then the fractal code's attractor is a fractal object. In fact, by iterating a determinist transformation on

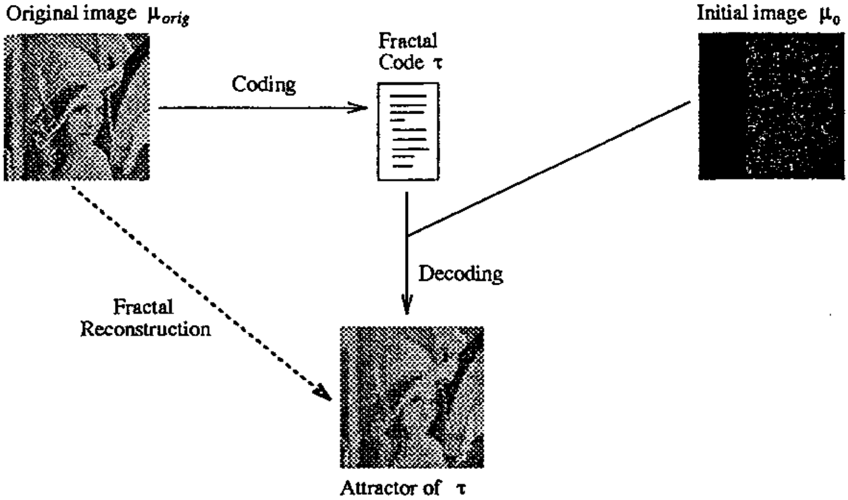


Fig. 2 Classical coding/decoding stage.

some initial image, a determinist fractal image is obtained. Therefore, this process must be independent of the resolution required. The original image μ_{orig} has a fixed size defined by the number of its pixels. But the fractal code τ has no intrinsic size, because it is a transformation which can be theoretically applied on any images.

Hence, we can assume that :

- The coding error is rather "small" (i.e. τ is a representative approximation of the original image μ_{orig}).
- Self-similarities (i.e. matchings between areas with different sizes, in the original image) are scale-independent (i.e. remain true at any level of the image's resolution).

Then, the fractal code enables to zoom. In practice, this operation (see Fig. 3) consists in increasing the range blocks' size, and therefore the domain blocks' size (because $D = 2 \cdot R$). For a zoom by a factor z, the new sizes will be $R' = z \cdot R$ and $D' = z \cdot D$ during the decoding stage, but the fractal code τ (i.e. all the local codes τ_i) will be unchanged.

A point which is not taken into account in this study is the coding error. As a matter of fact, the fractal coding is a lossy process, and the coding error is magnified during the decoding stage when zooming. So, a special treatment of this error would have been necessary to really compete with the classical interpolators.

5. IMPROVEMENTS USING OVERLAPPED RANGE BLOCKS

5.1 Introduction

When using the classical fractal zoom, the main problem is an important block effect (see Fig. 11(c)) i.e. some disturbing discontinuities along the range blocks' sides. Within the framework of the classical decoding process, some improvements have been done to perform a good visual quality. Emmanuel Reusens[4] uses overlapped range blocks instead of a partition and averages the two (or four) common areas of the range blocks.

In this case, the purpose is the fractal compression of a still image. That is why he must use a "weak" overlapping of the range blocks, in order to keep the compression aspect of

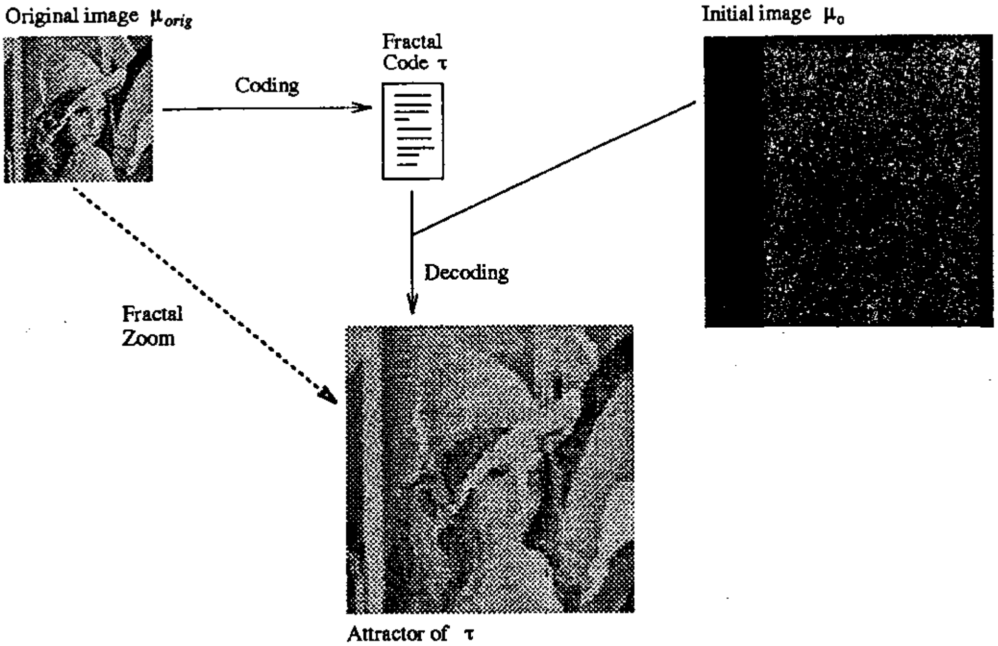


Fig. 3 Fractal zoom process

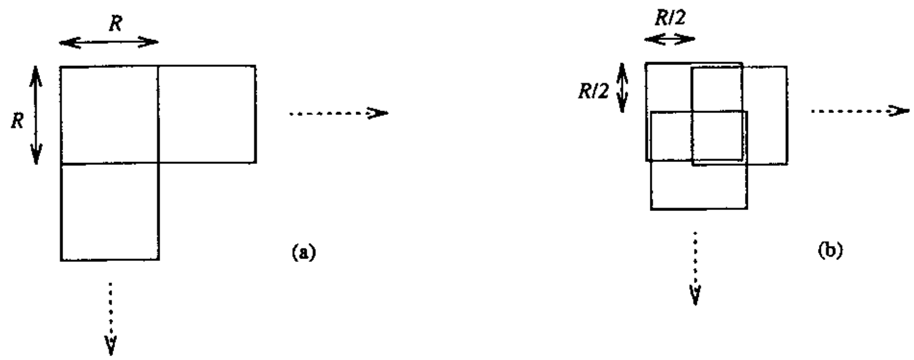


Fig. 4 Covering up of the image using (a) a partition and (b) overlapped range blocks with half of their side's size.

the method. As a part of oversampling, we do not care about compression and that allows us to keep an important redundancy of the information in the fractal code. In this study, the overlapping used is shown on Fig. 4(b).

The case described on figure 4(b) is equivalent to take four partitions $P^{(j)}$: $P^{(1)}$ of the entire image and the other partitions of a part of the image as shown on Fig. 5, with the same cells' size.

So, in order to encode the image μ_{orig}, we need to know the code of each part (see Fig. 5(a),(b),(c) and (d)) of the original image independently. Let us note $\tau^{(j)}$ the fractal code of the image $\mu_{orig}|_{P(j)}$ (i.e. the restriction of the image μ_{orig} to the partition $P^{(j)}$). Then, the fractal code of the image μ_{orig} is now defined by the formula :

$$\tau(\cdot) = \psi[\prod_{j=1}^{4} \tau^{(j)}(\cdot)] \qquad (2)$$

where ψ is a transformation used to "stick back" together the different codes $\tau^{(j)}$ (see Fig. 6) according to the original image.

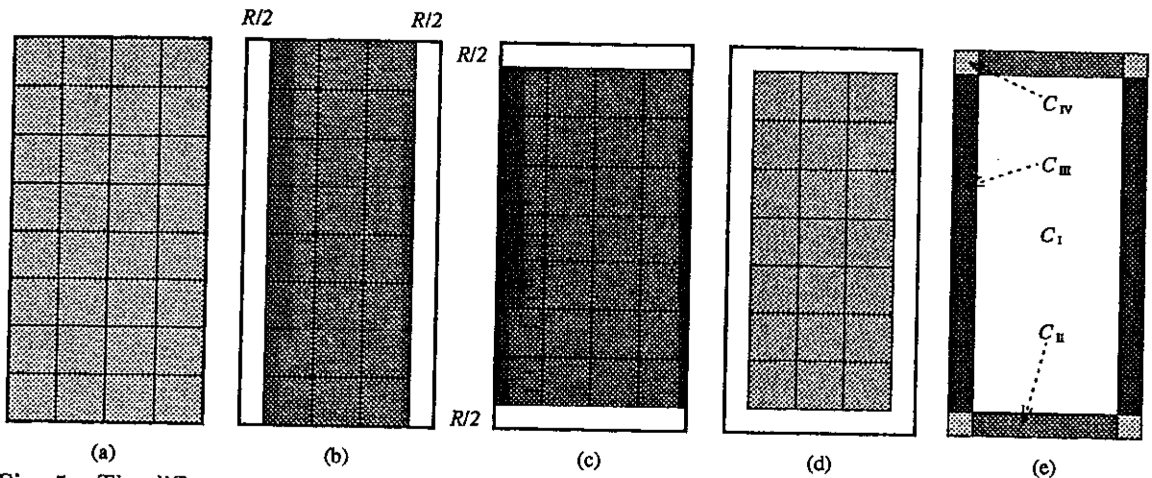


Fig. 5 The different areas considered in the image with the partition (a) $P^{(1)}$; (b) $P^{(2)}$; (c) $P^{(3)}$; (d) $P^{(4)}$ and (e) some suitable areas C_{I}, C_{II}, C_{III} and C_{IV}.

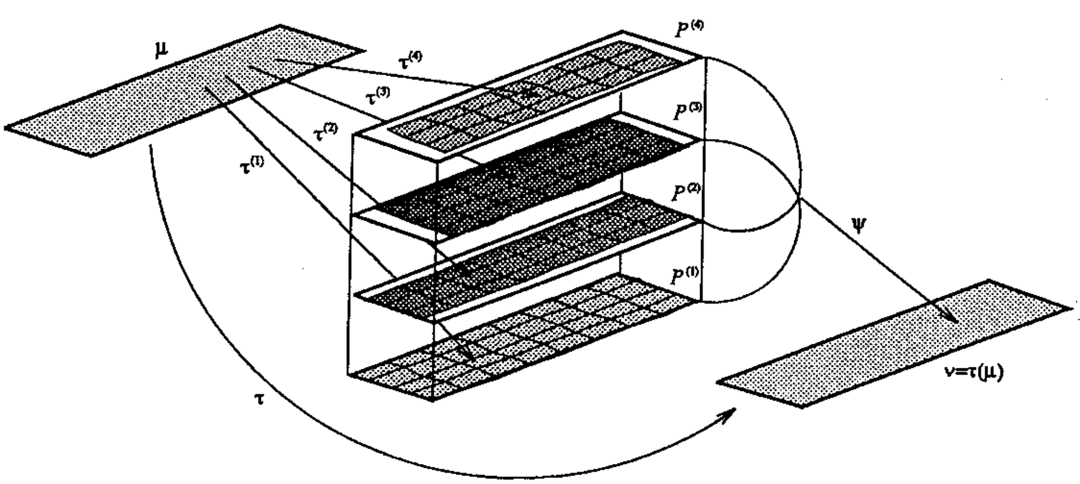


Fig. 6 Behavior of the new code τ and the transformation ψ.

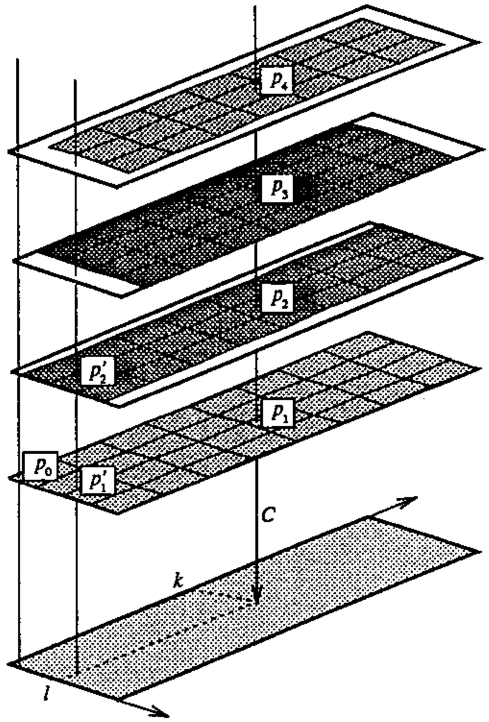


Fig. 7 Behavior of the function C.

To define completly the transformation ψ, we must include the image's sides processing. The image is divided into four areas' categories, as shown on the figure 5(e), in keeping with the figures 5(a),(b),(c) and (d). Let p be a pixel, (k,l) its coordinates and $V(p)$ the function which gives the value of the pixel p. Let us take five images with the *same size*: ν and, $\nu^{(j)}$ with the partition $P^{(j)}$ for $1 \leq j \leq 4$. Let us write $U = \prod_{j=1}^{4} \nu^{(j)}_{|_{P(j)}}$ and $C : U \longrightarrow \mathbb{R}^2 : p \longmapsto (k,l)$ the function which gives the coordinates, in the corresponding image, of a pixel belonging to U. In this case (Fig. 7), we obtain :

- $(k,l) \in C_{\text{I}} \Rightarrow C^{-1}(k,l) = \{p_1, p_2, p_3, p_4\}$
- $(k,l) \in C_{\text{II}} \cup C_{\text{III}} \Rightarrow C^{-1}(k,l) = \{p_1', p_2'\}$
- $(k,l) \in C_{\text{IV}} \Rightarrow C^{-1}(k,l) = \{p_0\}$

So, we can write :

$$\nu = \psi(U) = \bigcup_{p \in \nu^{(1)}} \zeta[C^{-1}(C(p))] \tag{3}$$

and ψ is given by the function ζ (which depends on the method used).

Therefore, in order to know the global code τ, we only have to take $\nu^{(j)}_{|_{P(j)}} = \tau^{(j)}(\mu)$ for some image μ, and then, using Eq. (2) and Eq. (3), the result is $\nu = \tau(\mu)$.

5.2 Fractal Zoom With Crushing (F.Z.C.)

This method shows how the overlapping of the range blocks can reduce by half the block effect. For example, it consists in putting each transformed domain block, by the corresponding $\tau^{(j)}$, without taking the others into account (see Fig. 8).

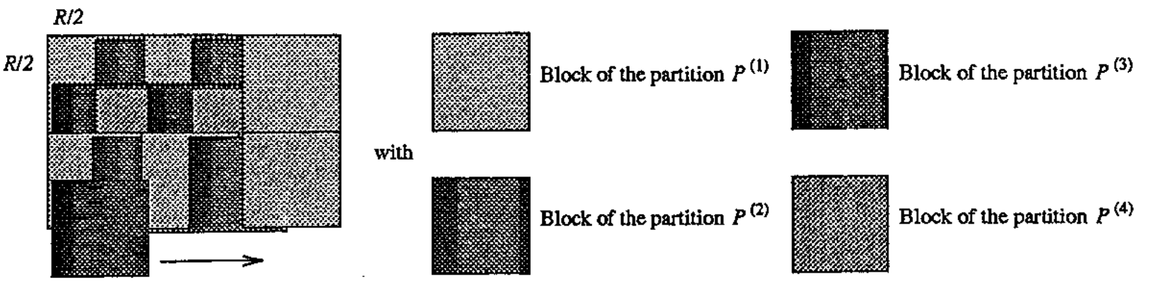


Fig. 8 "Crushing" of the range blocks.

5.3 Fractal Zoom With Total Averaging (F.Z.T.)

This method is the first and most natural improvement of the technique F.Z.C. It allows to take into account easily the redundancy of the coding. Most of the pixels are coded four times (area C_{I} of the image). So, in order to find the value of the transformed pixel, we only have to average the four values obtained with the codes $\tau^{(j)}$, $1 \leq j \leq 4$. In the area C_{II} (resp. C_{III}) the averaging is only done on the two values obtained with the codes $\tau^{(1)}$ and $\tau^{(2)}$ (resp. $\tau^{(1)}$ and $\tau^{(3)}$). In the area C_{IV}, we take the pixel obtained with $\tau^{(1)}$. Hence, ζ is defined by $V[\zeta(\{p_n\})] = \frac{1}{r} \cdot \sum_{p \in \{p_n\}} V(p)$ for r pixels $\{p_1, ..., p_r\}$ ($r \in \{1, 2, 4\}$) and $\zeta(\{p_n\})$ is a pixel belonging to ν such that the coordinates of $\zeta(\{p_n\})$ are equal to $C(p_1) = ... = C(p_r)$.

The main problem with this technique is that a "total averaging" smoothes the image too much. The following method preserves a sharpener aspect of images.

5.4 Fractal Zoom With Adaptive Averaging (F.Z.A.)

This method is a variant of the technique F.Z.T. based on the following consideration : if the fractal coding was perfect, the four independent codes $\tau^{(j)}$ would give the same results. But in practice, this seldom happens. So usually, we must choose between the four possible values given by the transformations $\tau^{(j)}$. There is no strict way to do this because, with the oversampling of an image, nothing about the original continuous bidimensional signal is known. Therefore, it is not possible to rigorously find the value of new sampled points. Our intuitive approach is the following : it is reasonable to suppose that the fractal coding has a rather stable behavior, and then, we have to rule out the "bad" artifacts of the method. Therefore, as the possible values should have been equal, it is reasonable to assume that the "best" ones are given by the two closest values.

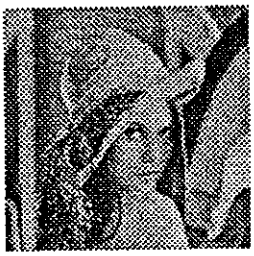
For example, if the possible values a. e 3, 7, 9 and 21, the choosen value will be $\frac{7+9}{2} = 8$, because 7 and 9 are the two closest values in the sequence $\{3, 7, 9, 21\}$. The value obtained with the method F.Z.T. would have given $\frac{3+7+9+21}{2} = 20$.

Hence, ζ is defined by :

- $(k, l) \in C_{\mathrm{IV}} \Rightarrow V[\zeta(p_0)] = V(p_0)$
- $(k, l) \in C_{\mathrm{II}} \cup C_{\mathrm{III}} \Rightarrow V[\zeta(p'_1, p'_2)] = \frac{V(p'_1)+V(p'_2)}{2}$

Thus, ζ is the same function as that of the previous sub-section, if the coordinates belong to $C_{\mathrm{II}} \cup C_{\mathrm{III}} \cup C_{\mathrm{IV}}$.

If the coordinates (k, l) belong to C_{I}, let us write $x_j = V(p_j)$ the value of the pixel p_j for $1 \leq j \leq 4$ and $d_{n,m} = |x_n - x_m|$. Therefore, the computation of six distances $d_{1,2}$, $d_{1,3}$,



(a)

(b)

Fig. 9 Image of Lenna : (a) λ_1 and (b) λ_2.

$d_{1,4}$, $d_{2,3}$, $d_{2,4}$ and $d_{3,4}$ (because $d_{n,m} = d_{m,n}$ and $d_{n,n} = 0$) is required. Then, there exists at least one couple (n_0, m_0) such that : $d_{n_0,m_0} = \min_{n<m}\{d_{n,m}\}$. Hence, let us define :

- $(k,l) \in C_{\mathrm{I}} \Rightarrow V[\zeta(p_1,p_2,p_3,p_4)] = \frac{x_{n_0}+x_{m_0}}{2}$.

In this case, ψ is an "adaptative averaging" which realizes the average computation from only two values, instead of four as seen in the previous sub-section.

6. COMPARISON BETWEEN FRACTAL ZOOMS AND CLASSICAL INTERPOLATORS

In this section, using three criteria, different oversamplers are compared : fractal zooms described previously and classical interpolators. Of course, those comparisons are difficult to establish because neither criterion is better than the other for a classification of the oversamplers and it depends on what we want to measure.

Let us note $\Delta(\cdot, z)$ an oversampling by a factor z with the method Δ, chosen from among the following ones : NNI, LI, CCI, FZ, FZC, FZA and FZT. Those techniques are tested on the images λ_1 which is the 128 by 128 image of the Lenna's face and λ_2 which is the 256 by 256 image of the Lenna's face as well (see Fig. 9).

Let us note $E_2(\cdot,\cdot) = [d_2(\cdot,\cdot)]^2$ the squared error and $e_2(\cdot,\cdot)$ the Mean Squared Error (M.S.E.).

6.1 M.S.E. Criterion

It is an analytical criterion which gives some information about the oversampler's quality. While this criterion is sometimes in contradiction with the visual quality of results, it allows an objective classification, counter to a visual one.

To work out this classification, we use the image λ_1 which is in this study the image λ_2 subsampled by a factor two by taking one pixel over four. Let us encode this image with $\tau(\lambda_1; 4, 2, 1, 1, 1.3)$ which gives the largest domain pool (set of the 8 by 8 domain blocks in this case). Let us write $e_\Delta = e_2(\lambda_2, \Delta(\lambda_1, 2))$. In this case, the reconstruction M.S.E.s are: $e_{\mathrm{NNI}} = 179$, $e_{\mathrm{LI}} = 76$, $e_{\mathrm{CCI}} = 75$, $e_{\mathrm{FZ}} = 217$, $e_{\mathrm{FZC}} = 216$, $e_{\mathrm{FZA}} = 172$ and $e_{\mathrm{FZT}} = 146$.

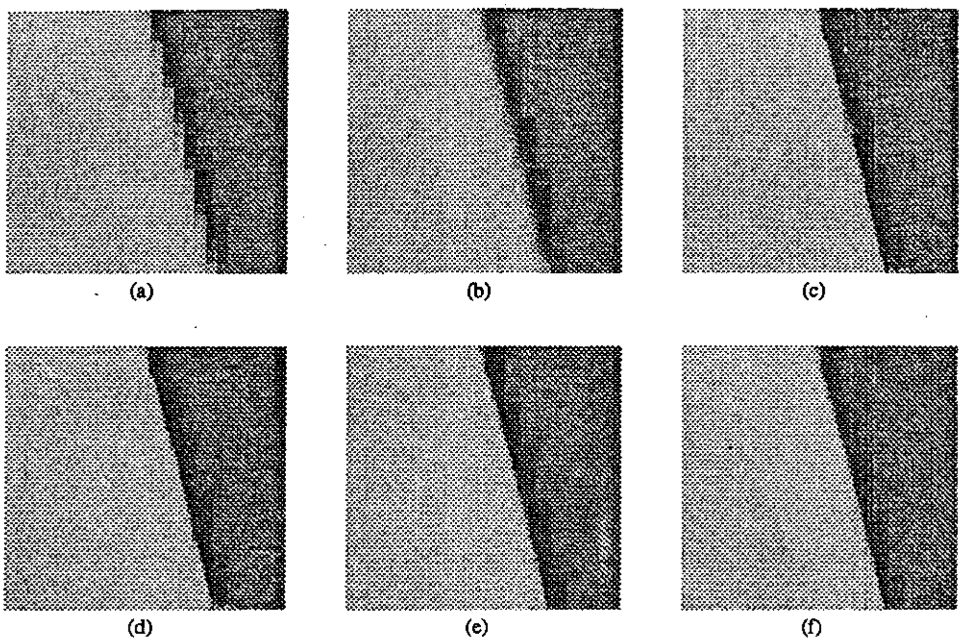


Fig. 10 Lenna's shoulder zoomed by a factor six with (a) Δ = NNI ; (b) Δ = LI ; (c) Δ = FZ ; (d) Δ = FZC ; (e) Δ = FZA and (f) Δ = FZT.

6.2 Visual Criterion

Using a zoom by a factor six of the image λ_2, some typical behaviors can be easily observed. The fractal codes are obtained with $\tau(\lambda_2; 4, 2, 1, 1, 1.3)$. Some interesting areas of the corresponding oversampled images are shown on Fig. 10 and Fig. 11.

6.3 "Original Luminance Values Respect" Criterion

This criterion is based on intuitive reasoning in order to validate an oversampler in another way. An optical subsampling can be well modeled, as opposed to oversampling, by a simple operation : an averaging. So, in order to show that the oversampler used is a good approximation of an optical zoom, the oversampling should be an inverse transformation of that subsampling.

Let us note $sub(\cdot, z)$ the subsampling, by a factor z, of an image by using a simple averaging. Let us take an original image μ_{orig}. Then, we can say that the oversampling must "respect the original luminance values". More precisely, if $\nu_{orig} = sub[\Delta(\mu_{orig}, z), z]$, we want that $\nu_{orig} \approx \mu_{orig}$. Let us write $\mu_{orig} = \{x_1, ..., x_k\}$ and $\nu_{orig} = \{y_1, ..., y_k\}$, where x_i (resp. y_i), $1 \leq i \leq k$, are the pixels' values of the image μ_{orig} (resp. ν_{orig}). Let us note $\phi_i = |x_i - y_i|$ for $1 \leq i \leq k$. Then, the distribution of ϕ_i must be concentrated near the value 0.

Of course, in this case, the technique N.N.I. is the best because $sub[\mathrm{NNI}(\mu, z), z] = \mu$ for all images μ and all zoom factors z (see Fig. 12(a)). But it is interesting to compare the other oversamplers with this criterion. The corresponding distributions are shown on Fig. 12, using the image $\mu_{orig} = \lambda_2$ and $z = 6$.

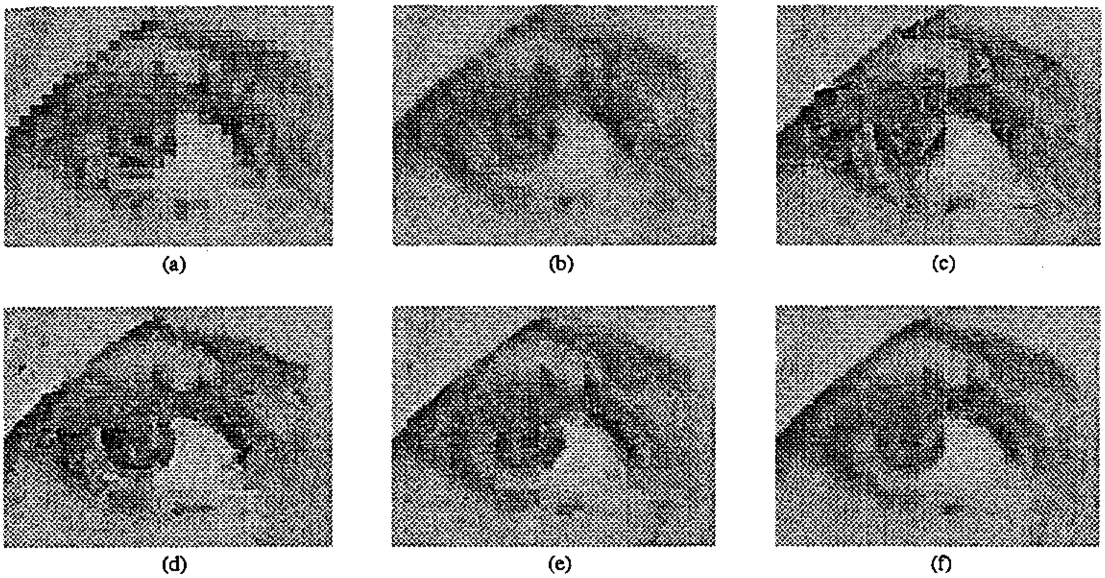


Fig. 11 Lenna's left eye zoomed by a factor six with (a) Δ = NNI ; (b) Δ = LI ; (c) Δ = FZ ; (d) Δ = FZC ; (e) Δ = FZA and (f) Δ = FZT.

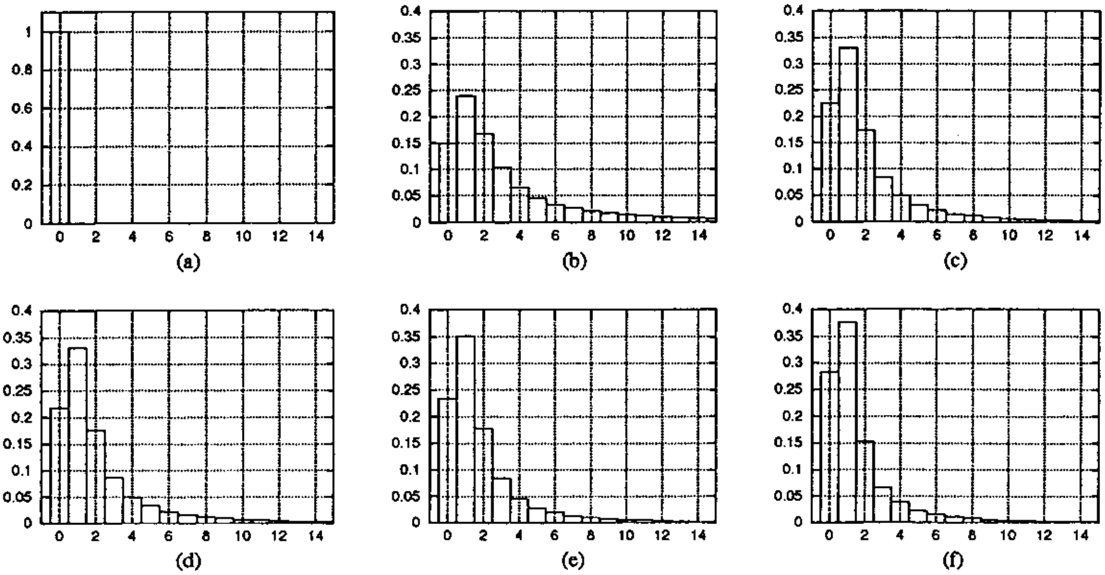


Fig. 12 Probabilties distribution of ϕ_i values with the methods (a) N.N.I ; (b) L.I. ; (c) F.Z. ; (d) F.Z.C. ; (e) F.Z.A. and (f) Z.F.T.

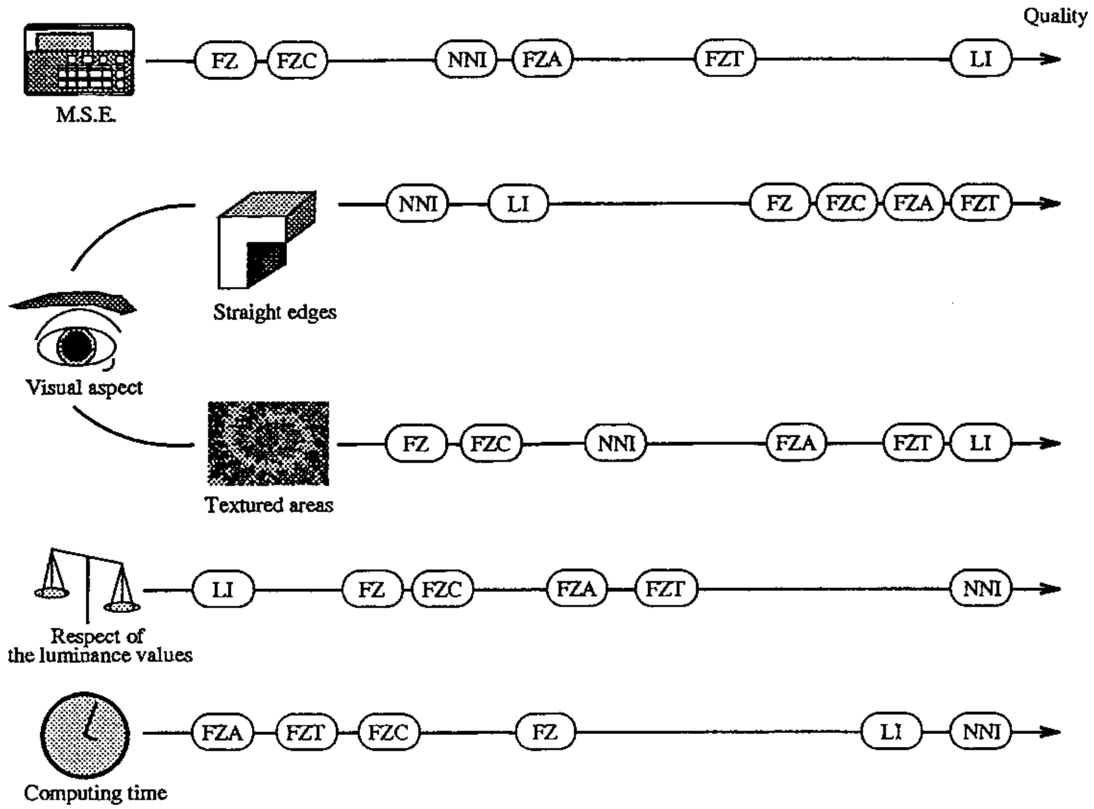


Fig. 13 Roughly done classifications of some oversamplers.

7. CONCLUSION

In this paper, after a review of image coding using I.F.S., we have presented how to use this coding in order to realize a zoom. A comparison with some spatial interpolators has shown that results obtained from the basic algorithm, used just as it is, are not better. Nevertheless, using some improvements to the classical method, we yield results on digitized images which show that the proposed versions put forth that I.F.S. (as an oversampler based on local spatial similarities) are an effective alternative to classical interpolators (based on spatial continuities) for image zooming.

But it is difficult to evaluate this oversampler versus others. As seen in section 6, several criteria can be used, and according to them results differ. Moreover, basic zoom can be defined from a coding scheme using I.F.S. Unfortunately, improvements of zoom functionality are done to the detriment of the compression aspect.

8. ACKNOWLEDGMENTS

This work is in part supported by AEROSPATIALE[5] (Etablissement de Cannes, France). The authors thank Philippe Perez, from AEROSPATIALE, for helpful discussions. We also thank Uriel Frisch, from the Observatory of Nice, and Gérard Iooss, from the Non Linear Institute of Nice, for the interest they devoted to this study, and Florence Dubois, from the EURECOM Institute, for reviewing the contents of this paper.

REFERENCES

1. L. Polidori and J. Chorowicz, *ISPRS J. Photo. Remote Sensing* **48**, 18 (1993).
2. R.G. Keys, *IEEE Trans. Acou. Speech Sig. Proc.* **29**, 1153 (1981).
3. A. E. Jacquin, *IEEE Trans. Imag. Proc.* **1**, 18 (1992).
4. E. Reusens, in *Proc ICASSP* (1994).
5. E. Polidori, *Zoom d'Images Fixes par I.F.S.* (in French) (Tech. Rep. 1-121, AEROSPATIALE/EURECOM, 1995).