# Using fractal compression scheme to embed a digital signature into an image

*Joan Puate, Fred Jordan*

Swiss federal institute of technology
Signal Processing Laboratory
CH-1015 Lausanne
Switzerland
Email: jordan@ltssg3.epfl.ch
Tel.: +41 21 693 70 89
FAX: +41 21 693 70 90

ABSTRACT

With the increase in the number of digital networks and recording devices, digital images appear to be a material, especially still images, whose ownership is widely threatened due to the availability of simple, rapid and perfect duplication and distribution means. It is in this context that several European projects are devoted to finding a technical solution which, as it applies to still images, introduces a code or Watermark into the image data itself. This Watermark should not only allow one to determine the owner of the image, but also respect its quality and be difficult to remove. An additional requirement is that the code should be retrievable by the only mean of the protected information. In this paper, we propose a new scheme based on fractal coding and decoding. In general terms, a fractal coder exploits the spatial redundancy within the image by establishing a relationship between its different parts. We describe a way to use this relationship as a means of embedding a Watermark. Tests have been performed in order to measure the robustness of the technique against JPEG conversion and low pass filtering. In both cases, very promising results have been obtained.

**Keywords**: digital signature, watermarking, image, copyright protection, security, fractal compression, IFS (Iterated Function Systems), FVT (Fractal Vector Technique), compression, internet

# I. Introduction

## I.1. The context

The last decades have seen exceptional development in the field of multimedia systems. Hence, people's needs will probably become very dependent on this phenomenon, and in order that the true potentials of these systems be properly exploited, some security mechanisms for privacy and intellectual rights must be given.

In the case of the protection of still images, finding a general solution is particularly difficult. The increasing number of digital networks and recording devices makes it very easy to create, distribute and copy such material. For these reasons, the demand for technical solutions against piracy is rapidly increasing among creators of multimedia information.

### I.2. Several approaches

Different approaches have been already presented, they all intend to face mainly format conversions, low pass filtering and data compression. Whereas some works are based on the direct modification of the pixels' luminance [1,2,3,7], some others make use of a predicted coding scheme [5] or a JPEG compression scheme [6]. In [4], the mark is embedded in the LSB (Least Significant Bit) of the pixels' values and in [8], the use of a frequency modulation is done. Also, some techniques have been developed for video data, which is the case of [9].

### I.3. Our proposal

In the following we propose a novel approach to Watermarking, based on the fractal theory of iterated transformations. For an image to sign we will construct its 'fractal code' in such a way that the decoded one includes a signature. Therefore, the signing algorithm consists of a coding-decoding process and retrieving the signature will be performed as a fractal coder. A signature thus obtained will have the important characteristic of being undetectable without the appropriate key.

Through this paper, some general concepts related to fractal coding techniques will be first explained, followed by their practical applications to our coder and decoder. Afterwards, we will introduce the signing and retrieving techniques as well as some results, focusing mainly on their robustness against JPEG compression and Blurring (3x3 kernel) attack. Finally, new ideas and possible improvements to be performed will be discussed.

## II. Fractal Image Coding

### II.1. Some general concepts

The fractals theory has proved to be suitable in many fields and particularly interesting in various applications of image compression. First important advances are due to M. F. Barnsley who introduces for the first time the term of Iterated Function Systems (IFS) [10,11,12,13] based on the self-similarity of fractal sets. Barnsley's work assumes that many objects can be closely approximated by self-similarity objects that might be generated by use of IFS simple transformations. From this assumption, the IFS can be seen as a relationship between the whole image and its parts, thus exploiting the similarities that exist between an image and its smaller parts.

At that point, the main problem is how to find these transformations or, what is the same, how to define the IFS. There is, in fact, a version of the IFS theory, the Local Iterated Function Systems theory, that minimizes the problem by stating that the image parts do not need to resemble the whole image but it is sufficient for them to be similar to some other bigger parts in it.

It was Arnaud E. Jacquin who developed an algorithm to automate the way to find a set of transformations giving a good quality to the decoded images [14]. In Fractal coding methods based on Jacquin's work, the main idea is to take advantage of the fact that different parts of the image at different scales are similar. As a matter of fact, they are block-based algorithms that intend to approximate blocks of a determined size with contractive transformations applied on bigger blocks. However, in theory the shape of the segments to encode is not restricted.

## II.2. The basic theory

The main idea of a fractal based image coder is to determine a set of contractive transformations to approximate each block of the image (or a segment, in a more general sense), with a larger block.

Some basic aspects of the theory are given in the lines below (a clear and brief explanation can be found in [15] and [16] ) :

Let's consider a metric space (£ , d) where d is a given metric and £ might be the space of the digital images. We can talk of a contractive transformation ß : £∅£ , when :

$$d(\ß(x), \ß(y)) \leq s\,d(x, y), \quad x,y \in £, \quad 0 \leq s < 1$$

In this case, exists a point x* such that:

$$\ß(x^*) = x^*$$

$$\lim_{n \to +\infty} \ß^n(x) = x^* \ , \forall\ x \in £$$

This point is called a fixed point.

An IFS consists of a complete metric space (£ , d) and a number of contractive mappings $\ß_i$ defined on £. The fractal transformation associated with an IFS is defined as:

$$B(E) = \bigcup_{i=1}^{N} \ß_i(E)$$

where E is any element of the space of non-empty compact subsets of £.

If $\ß_i$ is contractive for every i, then B is contractive and there is a fixed point for which:

$$A = B(A) = \bigcup_{i=1}^{N} \ß_i(A)$$

and

$$\lim_{n \to +\infty} B^n(E) = A$$

A is called the *attractor* of IFS and the transformations are usually chosen to be affine.

Once B is determined, it is easy to get the decoded image by making use of the Contraction Mapping Theorem : the transformation B is applied iteratively on any initial image until the succession of images does not vary significantly.

However, given a set M, how to find a contractive transformation B such that its attractor A is close to M?

To answer this question we have to apply to the *Collage Theorem*:

For a set M and a contraction B with attractor A:

$$h(M,A) \leq \frac{h(M,B(M))}{1-s}$$

Where h is the *Hausdorff Distance.*

That is to say that we can guarantee that M and A will be sufficiently close if we can make M and B(M) close enough.

In terms of $ß_i$ , and combining the two following expressions:

$$B(M) \spadesuit M ; \qquad B(M) = \bigcup_{i=1}^{N} ß_i(M)$$

we get

$$\bigcup_{i=1}^{N} ß_i(M) \approx M$$

So, if we make a partition of M:

$$M = \bigcup_{i=1}^{N} m_i$$

then, $m_i$ can be closely approximated by applying a contractive affine transformation $ß_i$ on the whole M:

$$m_i = ß_i (M)$$

The theory of IFS was extended to Local IFS where each part of the image is approximated by applying a contractive affine transformation on another part of the image:

$$m_i = ß_i (D_i)$$

where $D_i$ is the bigger part from which $m_i$ is approximated.

# III. Algorithm description

The main idea to automate the searching of a Local IFS relies on a partition of the image in blocks of a fixed size, called *Range Blocks*. These blocks are then approximated from larger blocks, called *Domain Blocks*. The transformations normally applied on the Domain Blocks are contracting and luminance scaling and shifting. Some other isometric transformations are sometimes used.

We have used an algorithm based on Jacquin's work as the first step of the signing technique. In the following, a brief explanation of it is given.

## III.1. Coder

Let O denote the image we want to encode. Let also $O_r$ denote a partition of O in *nxn* blocks referred to as *Range Blocks (Rb)*. Similarly, $O_d$ will denote another partition of O, this time in *2nx2n* blocks or *Domain Blocks (Db)* in steps of *nxn* pixels. The goal of the encoding algorithm is to establish a relationship between $O_r$ and $O_d$ in such a way that any Rb can be expressed as a set of transformations to be applied on a particular Db.

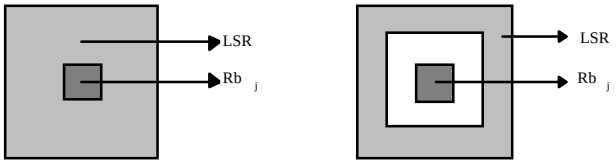
The transformations that have been considered are *Contraction*, *Isometric transformation* (one out of eight), *Luminance Scaling* and *Luminance Shifting*. For each Rb in O, denoted as $Rb_j$ , the code will consist of a vector $V_j$ and the appropriate transformations $T_j$, in such a way that:

- $V_j$ has its origin in $Rb_j$ and points to the correspondent $Db_j$ which now becomes its *Matching block ($Mb_j$ ).*

- $T_j$ if applied on $Mb_j$ , minimizes the Mean Square Error (MSE) with respect to $Rb_j$ .

- The couple $\{V_j, T_j\}$ is the best solution (in the sense of the MSE) within a local area surrounding $Rb_j$ in which we search for $Mb_j$.

The region of $O_d$ where the search of $Mb_j$ is performed is commonly taken as a square region surrounding the $Rb_j$. We will name this region LSR (Local Searching Region). The use of such a shape in the Matching Block determination might be justified from spatial redundancies considerations and that is essentially true. But that does not mean that other shapes can not give more than acceptable results on the Ranges Blocks approximation. Next figure shows the square surrounding region and a possible alternative:




*(a)* *(b)*

*Fig. 1: (a) A square LSR and (b) an alternative solution*

As we will describe further down, the assumption of this property will allow to make of this point the basis of our Watermarking proposal.

### III.2. Decoder

Let's consider an initial image *S* with the only constraint that it has to be of the same size as O. As before, we consider a nxn-partition $S_r$ in Rb, and a 2nx2n partition $S_d$ in Db. The decoding algorithm takes for each $Rb_j$ its Matching Block (pointed by $V_j$), applies on it the transformations (defined by $T_j$) and places the result back on $Rb_j$. These operations are performed for every $Rb_j$ of $S_r$ and through some iterations (typically four). After each iteration a new image $Q_i$ is obtained and it turns out that $Q_i$ converges to O, according to the *Collage Theorem*. An important point is that the solution $\{V_j , T_j\}$ obtained for O remains exactly the same for $Q_i$. That is to say that for every Range Block the same Matching Block as before is found. We will also take advantage of this point to embed the signature by a properly choice of the vectors $V_j$ .

Figure 2 shows some iterations for image Lena, when $S_0$ has been taken as a black image, n being equal to 4.

Figure 3, on the other hand, shows some iterations for image Lena, $S_0$ being a black image and n equal to 8.

It can be observed a better quality when n=4, above all in those parts of great detail. However, for n=4 the compression rate is much lower than for the case n=8. Therefore, there is choice to be made between quality and rate of compression. An intermediate solution might combine 4x4-Rangeblocks with 8x8-Rangeblocks. A quadtree based algorithm might achieve this compromise.

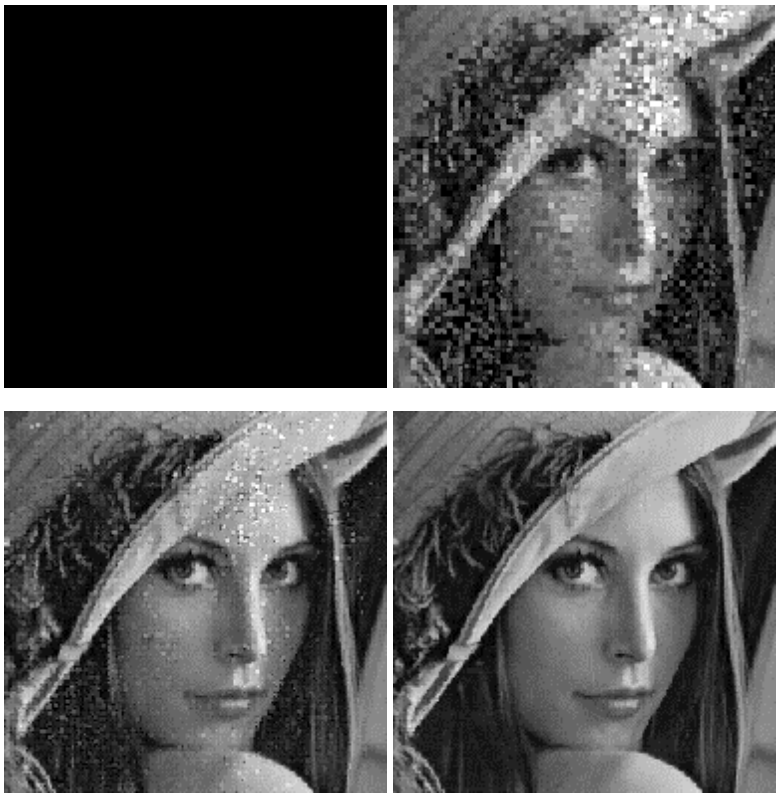


*Fig. 2. Iterations 1,2,4 of a code for "Lena" applied on a gray image (n=4).*

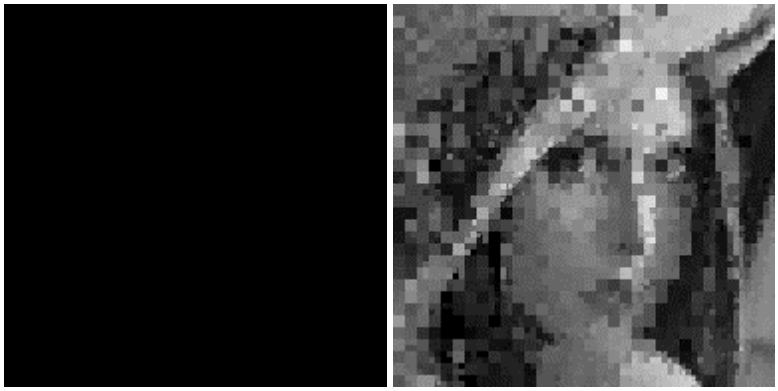


*Fig. 3. Iterations 1,2,4 of a code for "Lena" applied on a gray image (n=8).*

### III.3. Signer

Signing an image consists of a coding-decoding process with variable searching regions:

Let's consider two different LSR, A and B (Fig. 4), and a third one, C, defined as their union (a different choice of the regions could have been made, perhaps as a function of the characteristics of the image). Let also $S=\{s_0 , ... , s_{31} \}$ be a 32-bit signature. We will embed every bit with a redundancy U. The coding process is as follows:

- For each bit $s_i$ ,U Range Blocks are randomly chosen and denoted by $\{Rb\}_i$. The random function used to get the blocks makes use of a 'seed' that should only be known by the user.
- If $s_i = 1$, $\{Rb\}_i$ is coded by searching for $\{Mb\}_i$ in regions $\{A\}_i$ .
- If $s_i = 0$, $\{Rb\}_i$ is coded by searching for $\{Mb\}_i$ in regions $\{B\}_i$ .
- The rest of $Rb_j$ are coded by searching for $Mb_j$ in $\{C\}_i$. Note that this would be the case for all Range Blocks in a non-signed image.

Then the decoding is performed as described above. The resulting image contains the signature.

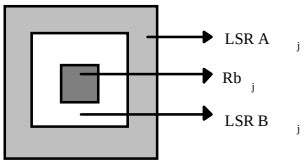



*Fig. 4: A range Block, its $LSR_A$ and its $LSR_b$. $LSR_c$ is defined as their union.*

### III.4. Retriever

The whole fractal code of an image can be expressed as the union of every Range Block single code:

$$m = \bigcup_j \{V_j, T_j\}$$

Likewise, for an image Q obtained after an iterative application of μ on any initial image, we consider its fractal code π . Since, in Q, every Range block is not just an approximation to a transformed Domain Block but it is exactly a transformed Domain Block, it turns out that $\pi = \mu$.

Thus, we will be able to identify the signature by simply accessing the Range Blocks of Q defined by the 'seed' used when signing, recoding them and checking the values of $V_j$ .

The rule to decide if a Range Block has been signed with a zero or a one, is the following one:

- If $V_j$ belongs to region $A_j$ , then a one has been embedded.
- If $V_j$ belongs to region $B_j$ , then a zero has been embedded.

In normal conditions, there ought to be a number of U recognition of bit one for those bits one in the signature, and of U recognition of bit zero for those bits zero in the signature.

To make the final decision there is a need of a threshold. It is going to be defined as the mean of the results obtained for bits two and three of the signature. Thus, they are always being embedded as a one and a zero, respectively.

## IV. Results

This section is divided in two parts. First one concerns the case for n=4, and second one the case for n=8. In both, the robustness to JPEG compression and to low pass filtering (3x3-kernel blurring) is discussed, as well as the quality of the signed images against the original and the non signed but fractal encoded.

All tests have been performed by embedding a 32-bits signature in 'Lena' image' (256x256), then applying the retriever. The chosen signature has a value of one in even bits and zero in odd bits. The redundancy U is equal to 50 for the case n=4, and equal to 25 for the case n=8.

The LSRs have been defined as follows:

$LSR_A$ : $k=\pm 6$; $l=\pm 6$;

$LSR_B$ : $|k|\leq 5$; $|l|\leq 5$;

Where (k,l) are the coordinates of vector V.

Moreover, we have defined a parameter P as the ratio between the number of vectors found in region A and the redundancy U. Since each bit equal to one has been embedded in LSR A, parameter P will be equal to one for those bits. Parameter P will be equal to 0 for bits equal to 0 (which have been embedded in LSR B).

Finally, we need to have a new parameter to express the reliability of the retrieved signature. Let's name it by σ:

$$\sigma(\%) = \frac{\sum_i |P_i - \xi|}{l * \xi} * 100, i = 0,...,l$$

where $\xi$ is the threshold value and l is the length of the signature.

### IV.1. Case n=4

*Figure 5* shows, for a 'n' equal to 4, the original image 'Lena', the decoded image of 'Lena' with no signature, and the decoded image of 'Lena' with the signature. Both, the Peak to Signal Noise Ratio (PSNR) between original and signed image and that between original and non-signed image present a value higher than 31.5 dB.

*(a)* *(b)* *(c)*

*Fig. 5 : (a) Original 'Lena' image; (b) decoded image of 'Lena' with no signature; (c) decoded image of 'Lena' with the signature (n=4)*

We have tested the robustness against JPEG compression qualities of 90, 75 and 50 %. Table I shows the results:

| | *No JPEG* | *JPEG 90%* | *JPEG 75%* | *JPEG 50%* |
|---|---|---|---|---|
| *P mean (even bits)* | 0.985 | 0.672 | 0.457 | 0.351 |
| *P mean (odd bits)* | 0.00 | 0.099 | 0.18 | 0.219 |
| *Threshold* $\xi$ | 0.5 | 0.39 | 0.34 | 0.31 |
| *Reliability* $\sigma$ *(%)* | 98.5 | 73.5 | 40.8 | 21.4 |
| *Bits correctly retrieved* | 32 | 32 | 32 | 29 |

*Table I*

### IV.2. Case n=8

*Figure 6* shows the original image 'Lena', the decoded image of 'Lena' with no signature, and the decoded image of 'Lena' with the signature. The PSNR between the original image and the non signed image presents a value of 25.82 dB (eighth iteration) whereas that between the original one and the signed decoded is equal to 25.40 dB (eighth iteration).

*(a)* *(b)* *(c)*

*Fig. 6 : (a) Original 'Lena' image; (b) decoded image of 'Lena' with no signature; (c) decoded image of 'Lena' with the signature (n=8)*

As before, we show in next table the behavior against JPEG compression. The same rates of quality have been tested:

| | *No JPEG* | *JPEG 90%* | *JPEG 75%* | *JPEG 50%* |
|---|---|---|---|---|
| *P mean (even bits)* | 0.992 | 0.962 | 0.887 | 0.797 |
| *P mean (odd bits)* | 0.00 | 0.002 | 0.035 | 0.097 |
| *Threshold* $\xi$ | 0.5 | 0.52 | 0.48 | 0.46 |
| *Reliability* $\sigma$ *(%)* | 99.25 | 92.3 | 88.8 | 77.2 |
| *Bits correctly retrieved* | 32 | 32 | 32 | 32 |

*Table II*

Table III shows the results when we have performed the tests of Table II but with a previous blurring on the signed image.

| | *No JPEG* | *JPEG 90%* | *JPEG 75%* |
|---|---|---|---|
| *P mean (even bits)* | 0.662 | 0.642 | 0.567 |
| *P mean (odd bits)* | 0.105 | 0.085 | 0.145 |
| *Threshold* $\xi$ | 0.40 | 0.36 | 0.36 |
| *Reliability* $\sigma$ *(%)* | 69.7 | 77.4 | 62.1 |
| *Bits correctly retrieved* | 32 | 32 | 32 |

*Table III*

## V. Conclusions and Future Works

We have presented an algorithm which succeeds in making digital signature functionality by modifying fractal features of the image.

The presented results show a good behavior for JPEG compression when n = 4. In that case, the algorithm has proved to be able to retrieve correctly the signature up to a JPEG quality of 75 % with a reliability of 40.8 %. When the quality went down to a value of 50 %, the number of badly recognized bits of the signature was of only three, though the reliability was, in that case, of 21.4 % .

When n = 8, the robustness against JPEG compression has turned out to be pretty high even for a quality of 50 % (reliability equal to 77.2 %). The method would have probably proved to be robust to higher compression rates though at these stages JPEG images may become damaged.

Concerning low pass filtering, the tests that have been performed for n = 4, showed some weakness against blurring convolutions. But for n = 8, the technique appeared to be very robust, even when the blurring attack was followed by a JPEG compression . When the compression rate was of 75%, we were able to retrieve the signature with a good reliability ($\sigma$ = 62.1 %).

For the case n = 4, the quality of the decoded images may be sufficient for some applications. However, the low complexity of the technique suggests that this quality can still be improved. On the other hand, new improvements ought to combine both cases, n=4 and n=8, in order to get either a good robustness against JPEG compression and low pass filtering and an acceptable level of quality. A quadtree-based algorithm seems promising as a mean to achieve this compromise. Indeed, a more advanced version of the actual work should take into account the statistics of the blocks where the signature is embedded.

A feature of this technique is that it does not allow, once the image has been decoded, to find out the location of the signature (in fact, it does not allow to determine whether a signature has been embedded or not). Since a Local Iterated Function Systems based algorithm looks for a set of transformations able to give a good approximation to the image to encode, it does not matter what these transformations are if the approximation is good enough. What the algorithm does, in fact, is to distinguish between different set of transformations by giving to one of them the feature of constituting a signature. Related to the last point would be the fact that we let totally opened the choice of searching regions shapes. The optimization of these shapes might increase the robustness of the signature as well as the retrieving reliability.

We think also that the *Fractal Vector Technique (FVT)* might be suitable in systems where images broadcasting needs a rate of compression similar to the one given by the fractal coding. Experiences have not shown great differences in quality between the decoded images either containing a signature or not. In effect, the degradation of the images comes mainly from the nature of the fractal method, not from the introduction of a watermark.

Finally, fractal compression scheme extracts several other parameters that might also be used to sign the image.

## VI. References


[1] J. Puate, F. Jordan, Two New Approaches to Digital Image Signature, *Diploma Project March 1996*.

[2] C. de Sola, F. Jordan, Enhancement of a watermarking algorithm in order to increase resistance to JPEG compression, *Diploma Project March 1995*.

[3] T. H. Kaskalis, I. Pitas, Applying Signatures on Digital Images.

[4] O. Bruyndonckx, J.-J Quisquater and B. Macq, Spatial Method for Copyright Labeling of Digital Images.

[5] Kineo Matsui, Kiyoshi Tanaka, Video-Steganography: How to Secretly Embed a Signature in a Picture.

[6] S. Burgett, E. Koch, S. Zhao, A Novel Method for Copyright Labeling Digitized Image Data.

[7] R. G. Van Schyndel, A Digital Watermark.

[8] Ingemar J. Cox, Joe Kilian, Tom Leighton, Talal Shamoon, Secure Spread Spectrum Watermarking for Multimedia, *NEC Research Institute, Technical Report 95 - 10.*

[9] T. Vynne, F. Jordan, Embedding a Digital Signature in a Video Sequence using Motion Vectors, *Submitted to ICIP 96.*

[10] M. F. Barnsley. *Iterated function systems.* In R. L. Devaney, L. Keen, K. T. Alligood, J. A. Yorke, M. F. Barnsley, B. Branner, J. Harrison, and P. J. Holmes, editors, Chaos and Fractals: The Mathematics Behind the Computer Graphics. American Mathematical Society, 1989.

[11] M. F. Barnsley, *Methods and apparatus for Image compression by iterated function systems.* United States Patent Number 4, 941, 193, 1990.

[12] M. F. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. *Proceedings of the Royal Society of London*, A399: 243-275, 1985

[13] M. F. Barnsley and L. P. Hud, *Fractal Image Compression*, AK Peters, Ltd., Wellesley, Massachusetts, 1993.

[14] Jacquin A., Image Coding Based on a fractal Theory of Iterated Contractive Image Transformations, *IEEE Transactions on image processing*, Vol1, pp 18-30, January 1992.

[15] L. Torres, M. Kunt, Video Coding, The second Generation Approach.

[16] Fisher Y., *Fractal Image Compression: Theory and Application*, Spinger Verlag Edition, New York, 1995.