A Parallel Approach to Image Decoding in the Fractal Image Block Coding Scheme

Ľubomír DEDERA – Ján CHMÚRNY
Department of Informatics and Computers, Military Academy,
031 19 Liptovský Mikuláš, Slovakia

In this paper a new parallel decoding algorithm for a fractal image coding scheme is designed. The algorithm utilizes a neural-network-like computational model where the number of steps corresponds to the number of iterations of the "traditional" iterative algorithm.

Keywords: Fractals; Image coding; Neural networks

1 Introduction

Fractal image compression, based on the theory of contractive transformations, began with the work of Michael Barnsley [1, 2] and Arnauld Jacquin [3, 4]. Since then a large amount of work on the topics has been undertaken which has given the fractal image coding the power to become a serious competitor of the established compression techniques. Neural networks (NN) were suggested for image decoding in fractal image coding schemes in [5] where two methods were introduced: the first method for generating Iterated Function System (IFS) attractors and the second method for generating invariant measures of IFS. Both proposed methods are of more theoretical than practical importance. The aim of this paper is to design a NN-like decoding algorithm for the "traditional" block-based fractal image coding scheme.

The rest of the paper is organized as follows: Section 2 describes the basic principles of the fractal image block coding. Section 3 introduces a new parallel decoding algorithm. The parallel computational structure the algorithm is based on has been derived from the mathematical model of the code which has been designed by means of the theory of graphs. This structure has much in common with recurrent neural networks. The algorithm itself utilizes the massive parallelism for the computation of the values of pixels. Section 4 summarizes the properties of the proposed parallel approach to image decoding.

2 Coding scheme

Let I be an original image. A set $\mathcal{R}$ of non-overlapping partitions of I, $\mathcal{R} = \{R_1, \ldots, R_n\}$ such that $\bigcup_{i=1}^{n} R_i = I$, is called a *range pool*; R_i, $i = 1, \ldots, n$ are called *range blocks*. In the experiments described later square range blocks of the same size have been used.

A set $\mathcal{D}$ of (overlapping) partitions of I, $\mathcal{D} = \{D_1, \ldots, D_m\}$, is called a *domain pool*; D_i, $i = 1, \ldots, m$ are called *domain blocks*. In this case square domain blocks of twice the size of range blocks distributed uniformly over the image I have been used.

Each domain block $D_i \in \mathcal{D}$ has been scaled down to the size of a range block using the pixel averaging operator **avg** and then 8 isometries of the square (ι_0-identity, ι_1, ι_2-reflections about mid-horisontal and mid-vertical axes, ι_3, ι_4-reflections about both diagonals, and $\iota_5, \iota_6, \iota_7$-rotations through 90°, 180°, 270°) have been applied on it [3, 6]. The resulting pool $\mathcal{C}$ of the size $8m$ is called a *codebook pool*.

Next we can consider each range block as a vector $R \in \mathrm{R}^p$ where p is the number of pixels in the range block R. The encoding problem for the range block R (using a codebook block C) is then the least squares problem

$$\min_{x \in \mathrm{R}^2} \|R - \mathbf{A}x\| \tag{1}$$

where $\mathbf{A}$ is a $p \times 2$ matrix with columns $C, (1, \ldots, 1)^T$, $x = (a, b)^T \in \mathrm{R}^2$ is a vector of coefficients, and the symbol $\|\cdot\|$ denotes the Euclidean norm defined by the formula

$$\|y\| = \sqrt{\sum_{i=1}^{p} y_i^2}. \tag{2}$$

The goal of the solution to (1) is to find the best approximation of the coded range block R using the codebook block C and the fixed basis block $(1, \ldots, 1)^T$. The results are two coefficients a, b, which can be interpreted as contrast scaling and luminance shift, for which (1) reaches the minimal value.

If the codebook block $C \in \mathcal{C}$ is not in the linear span $[(1, \ldots, 1)^T]$, then the minimization problem (1) has a unique solution

$$\bar{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T R \tag{3}$$

where the matrix $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T = \mathbf{A}^+$ is a so called *pseudo-inverse* of $\mathbf{A}$. The approximated "collage block" can be expressed as

$$\mathrm{col}(R) = \mathbf{A}\mathbf{A}^+ R. \tag{4}$$

The aim of encoding is to find the best-matching $C_R \in \mathcal{C}$ to each $R \in \mathcal{R}$ (with the lowest value $\|R - \mathbf{A}x\|$ among all $C \in \mathcal{C}$) and the corresponding coefficients a_R, b_R (1). The code of each range block then consists of the following items (see also Figure 1):

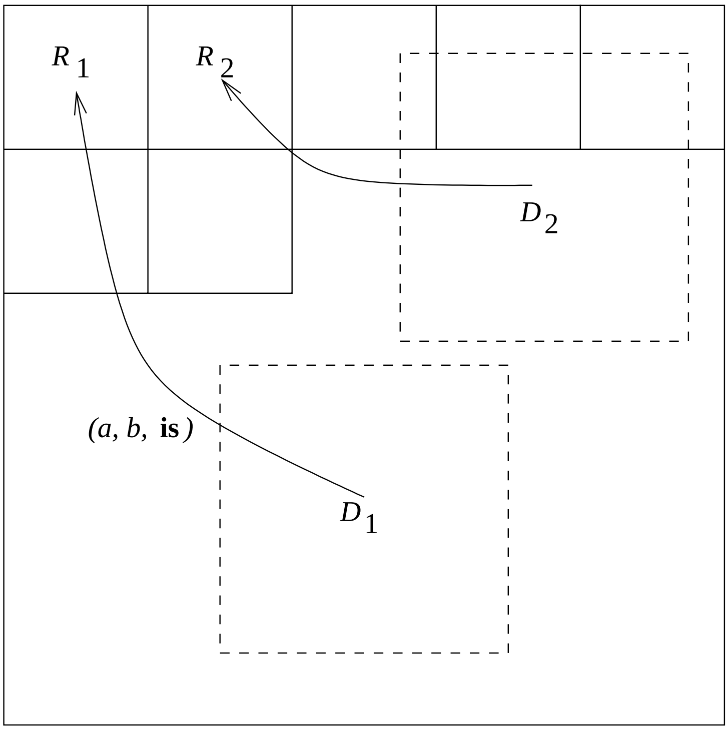

Figure 1: Description of the code

- index of the optimal domain block (corresponding to the optimal codebook block C_R)
- applied isometry $\mathbf{is}_R$ (corresponding to the optimal codebook block C_R)
- coefficient a_R, $|a_R| < 1$
- coefficient b_R.

Since the method is based on the theory of contractive transformations and Banach's fixed point theorem (the condition $|a_R| < 1$ assures convergence), *decoding* can be performed using the following iteration scheme

$$R^{(i+1)} = a_R\, \mathbf{is}_R \circ \mathbf{avg}(D_R^{(i)}) + b_R \tag{5}$$

where $\mathbf{is}_R$ denotes the isometry operator used for the coding of the range block R, **avg** the averaging operator applied on the optimal domain block D_R and equalizing the size of domain and range blocks, a_R the contrast scaling coefficient, b_R the luminance shift coefficient (both coefficients correspond to the particular range block R), and $^{(\dots)}$ denotes an iteration step (see also Figure 1). One iteration cycle consists of the application of eq. (5) on each $R \in \mathcal{R}$.

3 Parallel decoding algorithm

3.1 The mathematical model of the code

The mathematical model of the code can be described by means of directed graphs. Suppose, for simplicity, that the size of domain blocks is twice the size of range blocks. Then, the dependence of the pixel values of codebook (range) blocks (R) on the pixel values of domain blocks (D) is as depicted in Figure 2. In this case the value of a pixel is approximated using the value of four corresponding pixels in the domain block.

Let I be an original image, $\mathrm{R} = \{R_1, \dots, R_n\}$ its range partition, $\mathbf{D}$ an arbitrary domain pool consisting of blocks of twice the size of the size of range blocks in R and let C_I be the code of the image I defined by the formula

$$C_I = \{(R_i, D_i, \mathbf{is}_i, a_i, b_i), 1 \le i \le n\} \tag{6}$$

where R_i is the coded range block, $D_i \in \mathcal{D}$ is the corresponding domain block, $\mathbf{is}_i \in \{\iota_0, \dots, \iota_7\}$ is the used isometry, $a_i \in \mathbf{R}$ is the used contrast scaling and $b_i \in \mathbf{R}$ is the used luminance shift.

For the code C_I it is possible to construct the following directed graph G_{C_I} (Figure 3): $G_{C_I} = (V, H)$ where the set of vertices $V = V_1 \bigcup V_2$ contains vertices

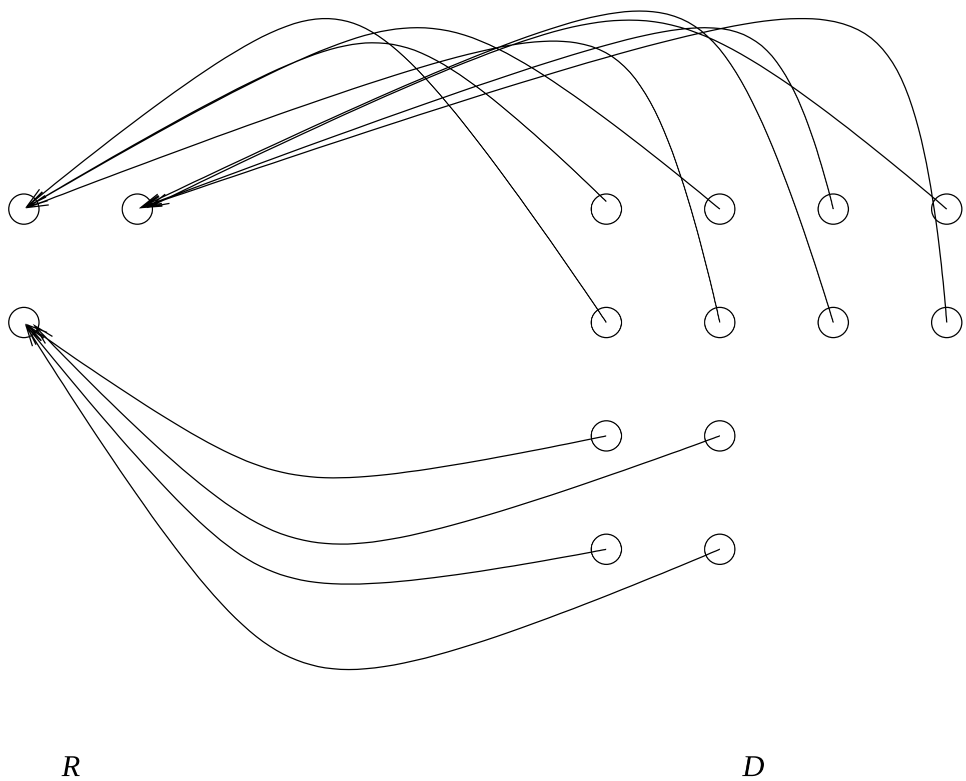

Figure 2: Correspondence of pixels

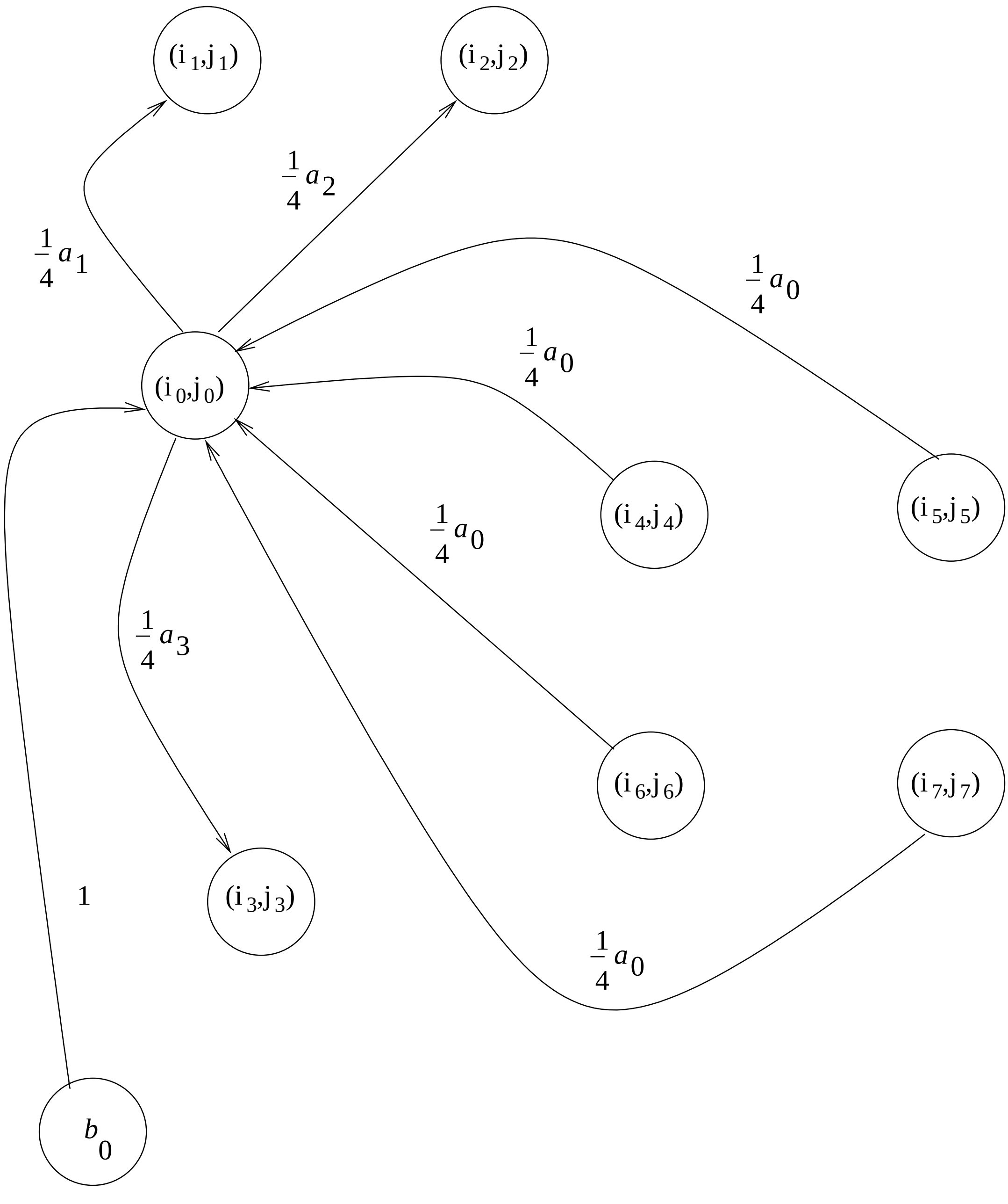

Figure 3: The graph representation of the code

of two types and the set of edges $H = H_1 \bigcup H_2$ contains edges of two types as well. The set of vertices V_1 contains all pixels, i.e.

$$V_1 = \{(i, j) : \ 0 \le i < m, 0 \le j < n\}. \tag{7}$$

The set V_2 contains vertices corresponding to the elements of the code C_I, i.e.

$$V_2 = \{x : x \in C_I\}. \tag{8}$$

From the point of view of interpretation the values of luminance shifts are used. The edges belonging to the set H_1 express the dependence of the value of each pixel on the values of four different pixels, i.e.

$$H_1 = \{[x, y] : y = (i, j) \in V_1 \wedge (\exists (R, D, \mathrm{iz}, a, b) \in C_I : y \in R \wedge x \in D_R(\mathrm{iz}(D), i, j))\} \tag{9}$$

where $D_R(D, i, j)$ denotes the set of those four pixels belonging to the domain block D which correspond (after pixel averaging) to the pixel $(i, j) \in R$ (Figure 2). An edge $[x, y] \in H_2$ if and only if the value of pixel $y \in V_1$ has been approximated with the quintuple corresponding to the vertex $x \in V_2$, i.e.

$$H_2 = \{[x, y] : \ y \in V_1 \wedge x \in V_2 \wedge y \in \mathrm{h}_R(x)\} \tag{10}$$

where h_R represents the projection $\mathrm{R} \times \mathbf{D} \times \{\iota_0, \dots, \iota_7\} \times \mathbf{R} \times \mathbf{R} \to \mathrm{R}$ defined by the formula $\mathrm{h}_R((R, D, \mathrm{iz}, a, b)) = R$. All the edges belonging to the set H_1 will be weighted with one quarter of the value of the corresponding coefficient a (contrast scaling), i.e. with the function $h_1 : \ H_1 \to \mathbf{R}$ defined by the formula

$$h_1([x, y]) = \frac{1}{4} a \iff \exists (R, D, \mathrm{iz}, a, b) \in C_I : \ y = (i, j) \in R \wedge x \in D_R(\mathrm{iz}(D), i, j) \tag{11}$$

and the edges in H_2 will be weighted with the constant 1, i.e. with the function $h_2 : \ H_2 \to \mathbf{R}$,

$$h_2([x, y]) = 1. \tag{12}$$

3.2 Parallel computational model

According to the graph G_{C_I} it is possible to construct a neural-network-type parallel computational model solving the image decoding problem in the following way:

- Each vertex of the graph G_{C_I} is assigned one neuron (the neurons are denoted by the same symbols as the vertices of the graph).
- Each neuron which corresponds to a vertex in the set V_1 is assigned the corresponding pixel and the state of each neuron in the $t-$th step, $t = 0, 1, 2, \dots$ expresses the value of the pixel which was determined in a particular step.

- Each neuron which corresponds to a vertex in the set V_2 is assigned a constant input expressing the value of the luminance shift which the particular vertex represents.

- The matrix of connection weights $\mathbf{W}$ between neurons with the size $|V| \times |V|$ is defined in the following way:

$$\mathbf{w}_{v_1,v_2} = \begin{cases} h_1(v_1, v_2), & \text{if } [v_1, v_2] \in H_1; \\ 1, & \text{if } [v_1, v_2] \in H_2; \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Let $v_i(t)$ represent the state of the neuron v_i in the $t-$th step. Then the parallel computational model described above represents a discrete dynamical system where the dynamics is defined by the equation

$$v_i(t+1) = \sum_{j=1}^{|V|} \mathbf{w}_{v_j v_i} v_j(t) \tag{14}$$

where $v_i \in V_1$ and

$$v_k(t) = h_b(x_k) \tag{15}$$

where $v_k \in V_2$, x_k is the coding quintuple corresponding to the vertex v_k and h_b is a projection $\mathrm{R} \times \mathbf{D} \times \{\iota_0, \ldots, \iota_7\} \times \mathbf{R} \times \mathbf{R} \to \mathbf{R}$ defined by the formula $h_b((R, D, \mathrm{iz}, a, b)) = b$.

Since at most five synapses leads to each neuron the matrix $\mathbf{W}$ will be sparse. The decoding algorithm itself can be described in the following way:

1. $t := 0$;
2. $v(0) := 0$ for all $v \in V_1$;
3. $v(t+1) := \sum_{j=1}^{|V|} \mathbf{w}_{v_j v} v_j(t)$;
4. $t := t + 1$;
5. If $t \leq$ the number of iterations required then
   - Goto 3;

   else
   - Finish.

The proposed parallel computational structure performs decoding using the iterative algorithm (Section 2), but it utilizes the fact that the values of pixels can be computed in parallel during single iteration steps. The states of the neurons will converge towards the states which represent the values of pixels after

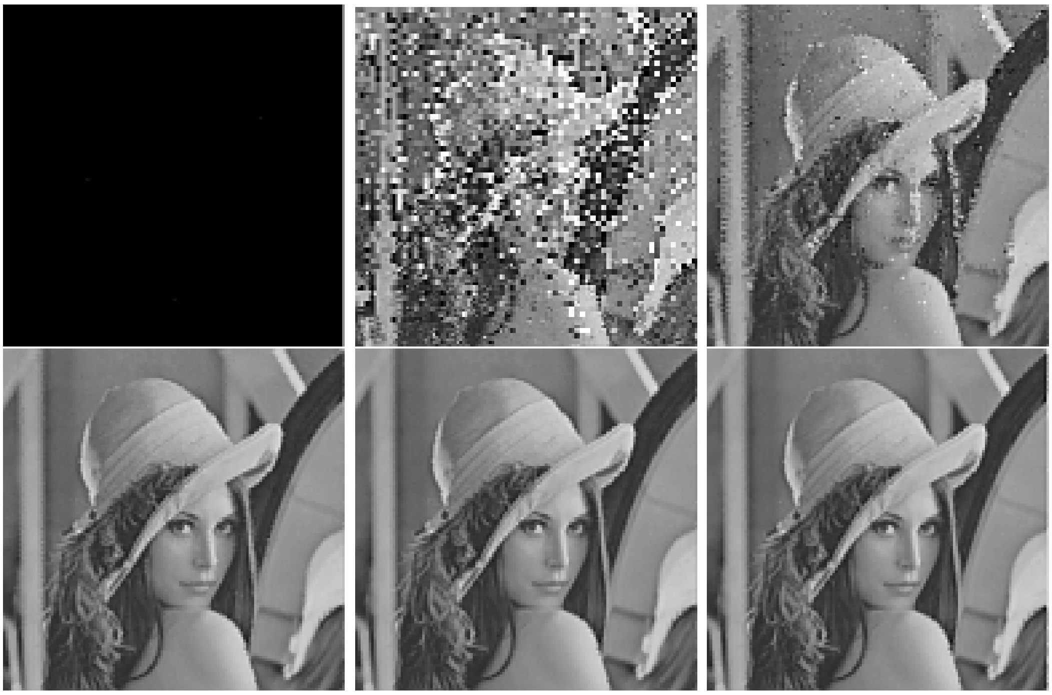

Figure 4: Results of decoding after 0, 1, 2, 3, 4, 5 iterations

decoding. The functional equivalence of both approaches (and the correctness of the new algorithm as well) comes from the fact that the four synapses leading to a neuron which corresponds to a pixel from neurons of the same type perform space contraction by pixel averaging and contrast scaling (multiplication by the coefficient a) at the same time. The synapse leading to the neuron from the neuron corresponding to a particular luminance shift b adds this value to the value described above.

The process of decoding for the 512x512x8 image "Lena" after the particular number of iterations can be observed from Figure 4. The image code has been obtained using range blocks of the size $8x8$, domain blocks of the size $16x16$ distributed uniformly over the whole image and a codebook of the size 1600 blocks. The bitrate achieved is 0.368 bpp. The process starts with a black image ($v_i(0) = 0$, $v_i \in V_1$) and the result after 5 iterations, in terms of peak-to-peak signal-to-noise ratio, is 30.50 dB.

4 Conclusion

The properties of the proposed parallel approach can be summarized as follows:

- The number of steps required for image decoding depends on the contrast scaling coefficients of the transformations used for image coding (practically 5 – 30).

- The process of learning in terms of neural networks is replaced by the image coding process (not performed by means of neural networks). The results of this process (6) are the connection weights (13), (11) of the designed parallel computational structure and the values of the neurons V_2 (15) which can be regarded as the input neurons.

- In the 3rd step of the algorithm only the values of neurons which belong to the set V_1 are computed since the states of neurons belonging to the set V_2 do not change their values.

- In the case when the computed values of pixels exceed the boundary values (e.g. 0 and 255), it is necessary to round them to these boundary values; naturally the same holds true for the basic iterative algorithm.

- Since the new parallel decoding algorithm and the basic iterative algorithm are based on the same theory, the results are also the same (e.g. [8, 9, 12]).

- The method can also be used in the case of a diffent ratio of the sizes of domain and range blocks; in this case it is necessary to modify the set of edges (synapses) H_1 and its weights in order to express the dependence of the values of pixels and averaging operations.

The proposed algorithm can be extended to other fractal coding schemes (e.g. [13]) with different shapes and sizes of range blocks.

References

[1] Barnsley, M. F., Sloan, A. D.: A better way to compress images, Byte No. 13(1988), pp. 215–223.

[2] Barnsley, M. F., Sloan, A. D.: Methods and apparatus for image compression by iterated function system, Patent 4,941,193, United States Patent Office, 1990.

[3] Jacquin, A. E.: A fractal theory of iterated Markov operators with applications to digital image coding, PhD thesis, Georgia Institute of Technology, Atlanta, August 1989.

[4] Jacquin, A. E.: Fractal image coding based on a theory of iterated contractive image transformations, SPIE Vol. 1360(1990), pp. 227–239.

[5] Bressloff, P. C., Stark, J.: Neural Networks, Learning Automata and Iterated Function Systems, in: Fractals and Chaos, Springer-Verlag, New York, 1990.

[6] Saupe, D., Hamzaoui, R.: Complexity reduction methods for fractal image compression, in: IMA Conf. Image Processing: Mathematical Methods and Applications, J. M. Blackledge (ed.), Oxford University Press, Oxford, 1995.

[7] Saupe, D.: Breaking the time complexity of fractal image compression, Technical report 53, Institut für Informatik, Freiburg, 1994.

[8] Lepsøy, S.: Attractor image compression – fast algorithms and comparisons to related techniques, PhD thesis, Norwegian Institute of Technology, Trondheim, 1993.

[9] Øien, G. E.: L_2-optimal attractor image coding with fast decoder convergence, PhD thesis, Norwegian Institute of Technology, Trondheim, 1993.

[10] Dudbridge, F.: Least-squares block coding by fractal functions. in: Fractal image compression – theory and applications to digital images, Y. Fisher (ed.), Springer-Verlag, New York, 1994.

[11] Bedford, T. J., Dekking, F. M., Keane, M. S.: Fractal image coding techniques and contraction operators, Technical Report, Delft University of Technology, Delft, the Netherlands, 1992.

[12] Dedera, Ľ.: Fractal image compression, PhD thesis, Military Academy, Liptovský Mikuláš, Slovakia, 1997 (in Slovak).

[13] Fisher, Y.: Fractal image compression with quadtrees, in: Fractal image compression – theory and applications to digital images, Y. Fisher (ed.), Springer-Verlag, New York, 1994.