Images Fractal Compression Optimization by Means of Artificial Kohonen Neural Networks

R.V. LOTOTSKIY

We considered the method of images fractal compression. The algorithm of clustering by means of artificial Kohonen neural networks was constructed. Results of work of the algorithm on real images at different parameters of the used neural networks are given.

Key words: images, fractal compression, optimization, Kohonen network, computational complexity.

Fractal compression of images distinguishes essentially from another known methods of compression with losses. The principal difference is that compression of information is not a result of unitary transformation of input data as, for example, in the case of Wavelet [1] or JPEG [2] compressions. The fractal compression represents coefficients of some contracting transformations for which an input image is a fixed point. For the first time the method of construction of such transformations for any image was developed by Barnsley [3].

The main problem of fractal compression is a computational complexity of work of the algorithm. To construct a contracting mapping, one is to transform and then compare a large quantity of parts or blocks of an image. For every block from some set, which is called the set of objective blocks, one looks for a block from another set, which is called the set of domain blocks, in such a way that the distance between two blocks would be minimum among all domain blocks from the corresponding set. A few methods of optimization of exhaustive search [4–9], which were based on clustering of the set of domain blocks, were suggested. But the majority of these methods either decrease lightly the computational complexity or result in high losses of the quality of an image. The method of clustering by means of artificial Kohonen neural self-optimizing network is least afflicted with these disadvantages [10]. For the first time, the Kohonen network was used for optimization of fractal compression by Bogdan and Meadows [11].

In this paper we construct the algorithm which enables to determine optimal domain blocks more accurately than it has been done in [11]. We have carried out comparative analysis of time of work of the developed algorithm and coefficients of compression with another known methods of fractal compression.

Let us cite the main definitions of the fractal theory which are necessary for understanding of further statement.

Definition 1. ([12]) Let (X, d) be a metric space. The transformation $T{:}X \to X$ is called the contracting mapping (or compression), if there exists the number s, $0 < s < 1$ such that

$$d(T(x), T(y)) \le s d(x, y), \quad x, y \in X. \tag{1}$$

The number s is called the coefficient of compression.

Theorem 1 ([12]). Any contracting mapping T defined in the complete metric space X has one and only one fixed point $x_0 \in X$:

$$T(x_0) = x_0.$$

ISSN 1064-2315

Definition 2. ([13]) Let X be a space of all nonempty compacts from $\mathbf{R}^2$ with the Hausdorff metric. A totality of the contracting mappings

$$W_1, W_2, \ldots, W_m, \quad W_i : X \to X,$$

with the coefficients of compression correspondingly $s_1 < 1, s_2 < 1, \ldots, s_m < 1$, along with a totality of the sets

$$E_0 \in X, \; E_1 = \mathbf{W}(E_0), \; E_2 = \mathbf{W}(E_1), \ldots, E_n = \mathbf{W}(E_{n-1}), \ldots,$$

where

$$\mathbf{W}(E_k) = \bigcup_{i=1}^{m} W_i(E_k). \tag{2}$$

we shall call the iterative functional scheme (IFS) acting in the metric space (X, d).

It may be proved that the space (X, d) is the complete one and the mapping $\mathbf{W}$ is contracting with the coefficient of compression $s = \max\{s_1, s_2, \ldots, s_m\}$ [13]. It follows from this that, as provided by theorem 1, there exists a single fixed point of the mapping $\mathbf{W}$.

IFS is a main instrument for construction of fractal sets.

While iterative construction every next set consists of m "diminished" copies of the previous set, and the sought-for fractal set is a fixed point of the contracting transformation $\mathbf{W}$. The initial set of such iterative process may be any nonempty compact from the corresponding space X.

The inverse problem is to find IFS for some given fractal set. The algorithm of IFS construction makes a search for subsets which are similar to the given fractal set, and constructs the corresponding contracting mappings.

Contracting transformations, which are built for images, have a structure similar to fractal IFS, and, consequently, this method has been called fractal.

The algorithm of fractal compression of images

Let us consider at first the mathematical model of an image which will be used for description of the algorithm.

Definition 3. The two-dimensional function

$$z = f(x, y), \; (x, y) \in \mathbf{I}^2, \; \mathbf{I} = [a, b] \subset \mathbf{R}, \; z \in \{z : z = j, \; j = \overline{0, N-1}\},$$

N is a number of levels of a gray color, we shall call the gray image or simply image.

To compute a distance between the images f and g we shall use the formula

$$d(f, g) = \left(\iint_{\mathbf{I}^2} | f(x, y) - g(x, y) |^2 dxdy \right)^{\frac{1}{2}}. \tag{3}$$

Here integration is carried out in the Lebesgue measure. Let us denote by $\mathbf{F}$ the space of all the image functions f with the metric d. It may be proved that (F, d) is a complete metric space, and, therefore, for any contracting transformation in this space there exists a single fixed point.

In practice one deals with real computer images and, therefore, instead of the image function $f(x, y)$ we shall consider a matrix of the values $[f(x_i, y_i)]$, $i = \overline{1, n}$, $j = \overline{1, m}$.. Correspondingly, formula (3) will get the form of a mean square error

$$d_{\text{discr}}(f, g) = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} (f(x_i, y_j) - g(x_i, y_j))^2 \right)^{\frac{1}{2}}.$$

Further, we shall call the points (x_i, y_j) the pixels as it is accepted in computer technology.

Let $f \in \mathbf{F}$ be a given image function. If there exists some contracting transformation $T_f(\bullet)$ on the space $\mathbf{F}$ such that f is a fixed point of this transformation, i.e.,

$$T_f(f) = f,$$

then to generate f with some accuracy ε, it will suffice to apply T_f certain times to any initial image g:

$$\exists n : d(T_f^{(n)}(g), f) < \varepsilon.$$

If for holding of coefficients of the constructed image smaller resources are used than for holding of f in an uncompressed form, than we shall get a compression of the initial image.

In the general case we shall look for such transformation $\mathbf{T}(\bullet)$ that for the given ε

$$d(\mathbf{T}(f), f) < \varepsilon.$$

The less is ε, the closer the fixed point $f_{\mathbf{T}}$ of the transformation $\mathbf{T}$ is to f. The following theorem determines this relationship between the distances $d(\mathbf{T}(f), f)$ and $d(f_{\mathbf{T}}, f)$.

Theorem 2 (theorem about collages [13]). Let f be some image. Assume, that for some contracting transformation $\mathbf{T}$

$$d(\mathbf{T}(f), f) \le \varepsilon,$$

then

$$d(f_{\mathbf{T}}, f) \le \frac{\varepsilon}{1-s},$$

where s is the compression coefficient $\mathbf{T}$, $f_{\mathbf{T}}$ is its fixed point.

As has been said, the structure of contracting transformations in the space $\mathbf{F}$ is similar to IFS for fractal sets. Since real images, as a rule, are not self-similar, instead of dividing an image into parts which are similar to a whole image, we shall look for similar parts of an image and construct the corresponding contracting transformations between these parts.

Let us construct some partition of an image into a set of the blocks R_i which we shall call the objective or range-blocks. Assign another block of an image D_i, which is called the domain block, and some affine transformation $\tilde{T}_i$ to every such block in such a way, that

$$R_i = \tilde{T}_i(D_i),$$

$$\bigcup R_i = \mathbf{I}^2,\ R_i \cap R_j = \varnothing, i \ne j,\ \bigcup D_i \subseteq \mathbf{I}^2, \quad (4)$$

$$\tilde{T}_i(x, y) = \mathbf{A}_i \begin{pmatrix} x \\ y \end{pmatrix} + \mathbf{b}_i,\ \ \mathbf{A}_i = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \text{ is a non-singular matrix.}$$

For every $\tilde{T}_i$ we shall determine the transformation in the space of images $\mathbf{F}$

$$T_i(f)(x, y) = c_i f(\tilde{T}_i^{-1}(x, y)) + o_i \ \ \forall (x, y) \in R_i. \quad (5)$$

This transformation maps the domain block D_i of the image f into the range-block R_i, changing at that the contrast range and brightness of a domain block with the use of the coefficients c_i and o_i correspondingly.

Theorem 3 ([7]). In order that the map T_i be contracting, it is sufficient the fulfillment of the condition

$$|c_i| \sqrt{|\det \mathbf{A}_i|} < 1, \tag{6}$$

where $\det \mathbf{A}_i$ is a determinant of the matrix $\mathbf{A}_i$, c_i is a coefficient of change in a contrast range.

Proof. By definition 1, T_i will be contracting, if

$$d(T_i(f|_{R_i}), T_i(g|_{R_i})) \le s d(f|_{D_i}, g|_{D_i}),$$

where $0 < s < 1$, $f|_{R_i}$ is a restriction of the function f on the set R_i.

Using (4), (5) and substituting the variables of integration, we have:

$$d^2(T_i(f|_{R_i}), T_i(g|_{R_i})) = \int_{R_i} |T_i(f)(x,y) - T_i(g)(x,y)|^2 dx\,dy =$$

$$= |c_i|^2 \int_{R_i} |f(\tilde{T}_i^{-1}(x,y)) - g(\tilde{T}_i^{-1}(x,y))|^2 dx\,dy =$$

$$= |c_i|^2 \int_{D_i} |f(u,v) - g(u,v)|^2 \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} du\,dv =$$

$$= |c_i|^2 |\det \mathbf{A}_i| \int_{D_i} |f(u,v) - g(u,v)|^2 du\,dv = |c_i|^2 |\det \mathbf{A}_i| d^2(f|_{D_i}, g|_{D_i}).$$

From this it follows condition (6).

The theorem is proved.

Let us define the transformation $\mathbf{T}: \mathbf{F} \to \mathbf{F}$:

$$\mathbf{T}(f)(x,y) = T_i(f)(x,y), \text{ if } (x,y) \in R_i,\ i = \overline{1,m}. \tag{7}$$

Since $\{R_i\}$ is a partition of $\mathbf{I}^2$, then $\mathbf{T}(f)$ will also be an image in the sense of definition 3. Since all the transformations T_i are contracting, $\mathbf{T}$ will also be contracting [13]. From here and by the theorem about a fixed point it follows, that $\mathbf{T}$ has a single fixed point $f_\mathbf{T} \in \mathbf{F}$:

$$\mathbf{T}(f_\mathbf{T}) = f_\mathbf{T}.$$

Let us consider the algorithm of construction of such range-partition at which the fixed point $f_\mathbf{T}$, of the corresponding transformation $\mathbf{T}$, is as close as possible to f. By means of ε we shall govern the coefficient of distorsion $f_\mathbf{T}$ relative to f.

Algorithm I ([3, 4])

1. The initial partition of an image into range-blocks.

Let us construct some partition of an image into the set of range-blocks **Range**:

$$\bigcup_{R_i \in \mathbf{Range}} R_i = \mathbf{I}^2,\ R_i \cap R_j = \varnothing,\ i \ne j.$$

Let us construct also the totality of domain blocks of different sizes **Dom:**

$$\bigcup_{D_j \in \mathbf{Dom}} D_j \subseteq \mathbf{I}^2.$$

The larger the number of domains in **Dom** is, the more possibilities there are to construct the "good" transformation **T**.

2. Search of a domain block, coefficients of the contrast range and brightness.

For every objective domain from the corresponding set we shall look for a domain area, affine transformation, the contrast range c_k and brightness o_k in such a way that the transformation T_k would be contracting and the condition

$$d(f(x,y), c_k f(\widetilde{T}_k^{-1}(x,y)) + o_k) < \varepsilon \quad \forall (x,y) \in R_k \in \mathbf{Range} \tag{8}$$

be valid.

If for some range-block R_k it is impossible to find the corresponding domain area, we shall divide this block into a few range-blocks of a less dimension and look for domain areas for each of them. It follows from theorem 2 that the less is ε, the closer to f will be the fixed point $f_{\mathbf{T}}$.

3. We shall continue the iterative process on stage 2 as long as for every objective block such domain area will be found for which condition (8) is fulfilled. For the case of computer images this partition will be necessarily completed, since for any range-block R_k of the dimension one pixel the coefficients of the contrast range and brightness can always be selected thus that (8) will hold true for any $\widetilde{T}_k$.

As a result of work of such algorithm, we shall obtain the necessary information for construction of the transformation $\mathbf{T}: \{\{R_i\}, \{\widetilde{T}_i\}, \{c_i\}, \{o_i\}\}$.

Search of optimal c_i, o_i in (8) for computer images we shall carry out so that to minimize the distance $d_{\text{discr}}(f(x_i, y_i), c_i f(\widetilde{T}_i^{-1}(x_i, y_i)) + o_i)$. Assume that a_l are values of the intensity of the pixels $f(\widetilde{T}_i^{-1}(R_i))$, b_l are values of the intensity of pixels of the objective block $f(R_i)$, and the number of pixels in the block R_i is equal to n. Then the values of the contrast range and brightness, which minimize the distance

$$E = \sum_{l=1}^{n} (b_l - (c_i a_l + o_i))^2, \tag{9}$$

will be the following:

$$c_i = \frac{n\left(\sum_{l=1}^{n} a_l b_l\right) - \left(\sum_{l=1}^{n} a_l\right)\left(\sum_{l=1}^{n} b_l\right)}{n\sum_{l=1}^{n} a_l^2 - \left(\sum_{l=1}^{n} a_l\right)^2}, \tag{10}$$

$$o_i = \frac{\sum_{l=1}^{n} b_l - c_i \sum_{l=1}^{n} a_l}{n}. \tag{11}$$

In algorithm 1, namely on step 2, partition of the input image into objective and domain blocks is not detailed. One of methods of such partition is a method of partition into squares. Let us consider it rather more detailed.

Ideally, we assume that the set **Dom** consists of all kinds square blocks of an image. In practical realizations, to simplify exhaustive search, the number of domains is noticeably less and is regulated, for example, by some step of "cutting" of image blocks. As it has been noted, the more there are such blocks, the more there are possibilities of construction of the "good" mapping **T**. Construction of the set **Range** is an iterative process. On the first stage we divide an input image into four equal squares (by means of addition of missing rows or columns of pixels, one can transform any image to the one of the dimension $2^k \times 2^k$) and record them in the set **Range**. For any recorded block we compute the distance between this block and every domain block from the set **Dom** using formula (8). Those range-blocks for which there exist domain blocks such that this distance is less then ε, and the condition of contracting transformation (6) is fulfilled, are unchanged. The rest are again divided into four squares, and the described procedure is repeated. Figure 1 shows an example of partition of an image into range-areas.

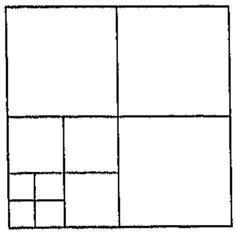

Figure 1

The compressed file has a block structure, at that every block contains the information which is necessary for construction of the transformation T_i.

Decompression, as it has been noted, is realized by means of recursive application of the transformation **T** to any input image which has the same size as the image which is to be restored.

Speed compression optimization by means of a self-organizing Kohonen network

A self-organizing Kohonen network is used for clustering of a set of vectors [7]. Figure 2 represents the sketchy scheme of such neural network. It consists of two levels. Vectors from some totality come in the first level, at that the number of neurons at the input level is not equal to the dimension of input vectors. The number of neurons at the second level is equal to the number of clusters this assemblage is to be divided into. Every output neuron is connected with some memory which, after work of the clustering algorithm, will contain all vectors from the corresponding cluster. The weight vectors $\mathbf{w}_i = (w_{ij})$ are varying during the work of the algorithm and after its termination will be close to the geometric centers of the corresponding clusters.

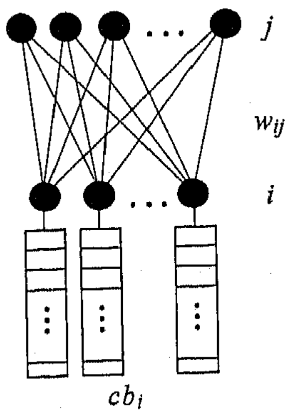

Figure 2

The considered Kohonen network enables to optimize the problem of search of the nearest domain block for some range-block by means of clustering of the totality of all domain blocks. The search of the nearest domain will be realized among domains of a few clusters instead of the totality of all domains.

Formulas (10), (11) define the optimal values of the contrast range and brightness for minimization of a distance between objective and domain blocks. If one uses a Kohonen network for domains classification, there is no necessity to optimize the contrast range and brightness — it is only necessary to determine a minimally possible distance between the range-block and domain block. The results of work [8] enable to construct the algorithm of search of minimum distances without finding the optimal values of contrast range and brightness.

Let us rewrite formula (9) in the vector form, assuming that $\mathbf{r}_i \in \mathbf{R}^n$ is a vector of the intensity of the range-block R_i, $\mathbf{d}_j \in \mathbf{R}^n$ is a vector of the intensity of the domain block D_j which is transformed to the size of the corresponding range-block:

$$E(\mathbf{r}_i,\mathbf{d}_j)=\min_{c,o}\|\mathbf{r}_i-(c\mathbf{d}_j+oC)\|,\quad c\in\mathbf{R},\ o\in\mathbf{R},$$

where $C\in\mathbf{R}^n, C=(1,...,1)/\sqrt{n}$. Let O be an operator of orthogonal projection which projects $\mathbf{R}^n$ on the orthogonal complement $\Omega^{\perp}$; Ω is a linear envelope of the vector C. For $Z=(z_1,...,z_n)\in\mathbf{R}^n\setminus\Omega$ we shall define the operator

$$\varphi(Z)=\frac{\mathrm{O}Z}{\|\mathrm{O}Z\|}.$$

Theorem 4 ([12]). Assume that $n\geq 2$ and $\mathrm{X}=\mathbf{R}^n\setminus\Omega$. Let us define the function Δ: $\mathrm{X}\times\mathrm{X}\to[0,\sqrt{2}]$ in the following way:

$$\Delta(\mathbf{d},\mathbf{r})=\min(\|\varphi(\mathbf{r})+\varphi(\mathbf{d})\|,\ \|\varphi(\mathbf{r})-\varphi(\mathbf{d})\|).$$

For $\mathbf{r}_i,\mathbf{d}_j\in\mathrm{X}$ the minimum distance $E(\mathbf{r}_i,\mathbf{d}_j)$ will be determined by the formula

$$E(\mathbf{r}_i,\ \mathbf{d}_j)=(\mathbf{r}_i,\ \varphi(\mathbf{r}_i))\,g(\Delta(\mathbf{r}_i,\ \mathbf{d}_j)),$$

where

$$g(\Delta)=\Delta\sqrt{1-\frac{\Delta^2}{4}}.$$

This theorem enables to minimize the distance $d_{\mathrm{discr}}(\varphi(\mathbf{r}_i),\pm\varphi(\mathbf{d}_j))$ instead of minimization of $E(\mathbf{r}_i,\mathbf{d}_j)$. A set of all the vectors $\pm\varphi(\mathbf{d}_j)$ we shall denote by $\mathbf{Dom}_{\varphi}$, and the set $\{\varphi(\mathbf{r}_i)\}$ — by $\mathbf{Range}_{\varphi}$. The vectors $\varphi(\mathbf{r}_i)$ we shall call the range-vectors, and $\varphi(\mathbf{d}_j)$ — the domain vectors, correspondingly.

Let us consider the Kohonen network algorithm as applied to our problem, namely, to clustering of domain vectors. In the beginning, vectors from domain (learning), and, later on, objective (choice of an optimal domain) sets will input the network.

Work of such network consists of two stages.

1. Training. A learning sample consists of the vectors $\{\pm\varphi(\mathbf{d}_j)\}$ of the same dimension. In what follows, the vectors dimension $\varphi(\mathbf{d}_j)$ we shall call the network dimension. After the stage of training, the domain vectors will be divided into subsets or clusters, the weight vectors $\mathbf{w}_i=(w_{i1},w_{i2},...,w_{im})$ being close to geometric centers of the corresponding clusters.

2. Choice of the nearest domain vector from a training sample for every objective vector.

The following algorithms for training and choice of the nearest domain are proposed.

Algorithm 2 (training of the network).

1. Network initialization.

We initialize in some way the weight vectors $\mathbf{w}_i$. One of sets of initializing elements may be some subset of domain vectors.

2. Search of the nearest weight vector for $\varphi(\mathbf{d}_j)\in\mathbf{Dom}_{\varphi}$.

From the learning sample $\mathbf{Dom}_{\varphi}$ we choose the element $\varphi(\mathbf{d}_j)$ and input it the network. Among the weight vectors $\mathbf{w}_i$ we find such vector $\mathbf{w}_k$ that

$$d_{\mathrm{discr}}(\mathbf{w}_k,\varphi(\mathbf{d}_j))\leq d_{\mathrm{discr}}(\mathbf{w}_i,\varphi(\mathbf{d}_j))\ \ \forall i,$$

and record the vector $\varphi(\mathbf{d}_j)$ into the corresponding memory cb_k.

3. Modification of the found weight vector.

We modify the weight vector $\mathbf{w}_k$ in the following way:

$$\mathbf{w}_k = \mathbf{w}_k + \gamma(\varphi(\mathbf{d}_j) - \mathbf{w}_k),\ 0 < \gamma < 1.$$

The parameter γ regulates the degree of an influence of the input vector upon changing of the corresponding weight vector. The rest weight vectors do not change. We remove the element $\varphi(\mathbf{d}_j)$ from the set $\mathbf{Dom}_\varphi$ and further, if the set $\mathbf{Dom}_\varphi$ is empty, terminate the algorithm, otherwise, return to item 2.

After fulfillment of this algorithm the weight vectors $\mathbf{w}_i$ will be close to the geometric centers of the corresponding clusters cb_i. The proof of this fact in the general form can be found, for example, in [10]. It should be observed that, for reduction of the computational complexity in this algorithm, only one training iteration is realized.

Algorithm 3 (choice of the nearest domain vector for the range-vector $\varphi(\mathbf{r})$).

In this algorithm three variables *cb_min*, *ind_dom_min*, *d_min* are used. After work of the algorithm they will contain the cluster index in which the nearest vector $\varphi(\mathbf{d}_j)$ is located, index of this vector in the found cluster and distance between this vector and $\varphi(\mathbf{r})$ correspondingly.

1. For every cluster cb_i we shall find the distance between each vector from this cluster and the corresponding "center" of this cluster $\mathbf{w}_i$. This distance we shall call the vector orbit:

$$o_{ij} = d_{\text{discr}}(\varphi(\mathbf{d}_{ij}), \mathbf{w}_i)\quad \forall \varphi(\mathbf{d}_{ij}) \in cb_i.$$

Let $o_{i\min} = \min_j o_{ij}$ and $o_{i\max} = \max_j o_{ij}$ be minimum and maximum orbits in the cluster cb_i correspondingly. Let us find the distance between the given range-vector $\varphi(\mathbf{r})$ and centers of every cluster cb_i:

$$d_i = d_{\text{discr}}(\varphi(\mathbf{r}), \mathbf{w}_i)\quad \forall \mathbf{w}_i.$$

Let us order the values d_i with respect to a magnitude. Let $d_{i_1}, d_{i_2}, \ldots, d_{i_m}$ be an ordered sequence. Let us initialize the variable-iterator l and *d_min*:

$$l := 1,\ d\_min = +\infty.$$

2. We shall find in the cluster cb_{i_l} the domain vector which is nearest to $\varphi(\mathbf{r})$. If the distance between this vector and $\varphi(\mathbf{r})$ is less than *d_min*, then we assign to *cb_min*, *ind_dom_min*, *d_min* the new values:

$$cb\_min := i_l,$$

$$ind\_dom\_min := \arg\min_j d(\varphi(\mathbf{d}_{i_l j}), \varphi(\mathbf{r})),$$

$$d\_min := \min_j d(\varphi(\mathbf{d}_{i_l j}), \varphi(\mathbf{r})).$$

3. $l := l + 1$.

4. If $l > m$, the algorithm is terminated. If $l \le m$ also for the vector cb_{i_l} the condition

$$d_{i_l} - o_{i_l\max} \le d\_min, \tag{12}$$

holds true, then we proceed to item 2, otherwise, proceed to item 3.

After work of this algorithm the nearest vector $\varphi(\mathbf{d}_{cb\_min,\, ind\_dom\_min})$ will be found. Condition (12) ensures that search in clusters, which certainly do not contain the sought nearest domain vector, will not be carried out.

The results of experimentation

Let us consider real examples of using algorithms 1–3. As a test example, we have chosen the most known image "Lenna" of the dimension 512×512. This image is most often used while demonstration of results of investigation in images processing and, therefore, enables to compare results of compression sufficiently objectively. The table gives the obtained results.

Table

Compression coefficient	PSNR, db	Time, sec	The minimum range-block	The maximum range-block
5.25	34.58	13	4.00	8.00
6.80	34.38	9	4.00	8.00
8.00	33.89	7	4.00	8.00
9.90	32.84	4	4.00	8.00
10.14	32.67	4	4.00	8.00
16.40	30.89	18	8.00	16.00
20.97	30.84	12	8.00	16.00
24.11	30.67	10	8.00	16.00
26.11	30.39	9	8.00	16.00
28.53	29.93	8	8.00	16.00
30.11	29.71	8	8.00	16.00
31.82	29.36	6	8.00	16.00
37.09	28.53	6	8.00	16.00
65.54	27.13	7	16.00	16.00
84.58	26.68	26	16.00	32.00
100.00	26.05	23	16.00	32.00
114.26	25.54	22	16.00	32.00
123.23	25.00	27	16.00	32.00

In the forth and fifth columns the minimum and maximum dimensions of range-blocks are given. These parameters regulate a number of the constructed networks, since a separate network will be built for every set of range-vectors of the same dimension. The table shows that increase of the minimum dimension of range-blocks results in considerable change of the compression coefficient and, correspondingly, the quality of a restored image (peak signal-to-noise ratio) — PSNR. It is obvious, the more "large" range-blocks there are in a compressed file, the better is the compression coefficient (and, correspondingly, the worse is the quality of a restored image). Increase of the minimum and maximum dimensions of blocks leads to increase of time of work of the algorithm, since training of the network which contains "larger" blocks engages more time than in the case of lesser blocks.

Figure 3 gives a dependence of PSNR upon the compression coefficient for algorithms 1–3 (*a*), median split clustering [9] (*b*), clustering with the use of Kohonen networks. These results can be found in [7] (*c*), Fisher clustering (*d*). The data are cited by [8].

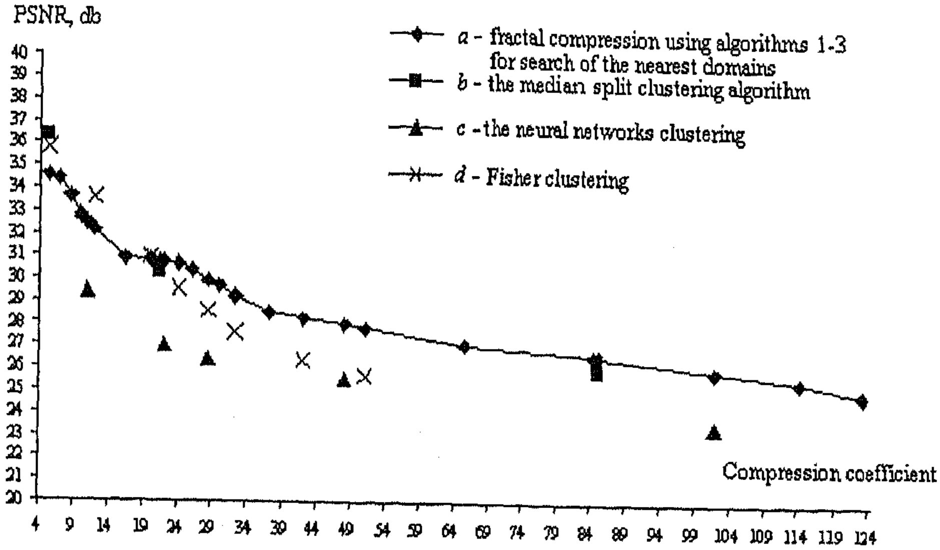

Figure 3

Unfortunately, exact time data were available only for the case of Fisher classification. Therefore, Figure 3 gives comparison of only two methods.

Figure 4 gives a dependence of time of work of the algorithm upon the compression coefficient for algorithms 1–3 (Pentium III, 800 MHz, 256 Mb RAM) (*a*), Fisher clustering (SGI Indigo2, R4400 processor) (*b*). These results are cited by [9].

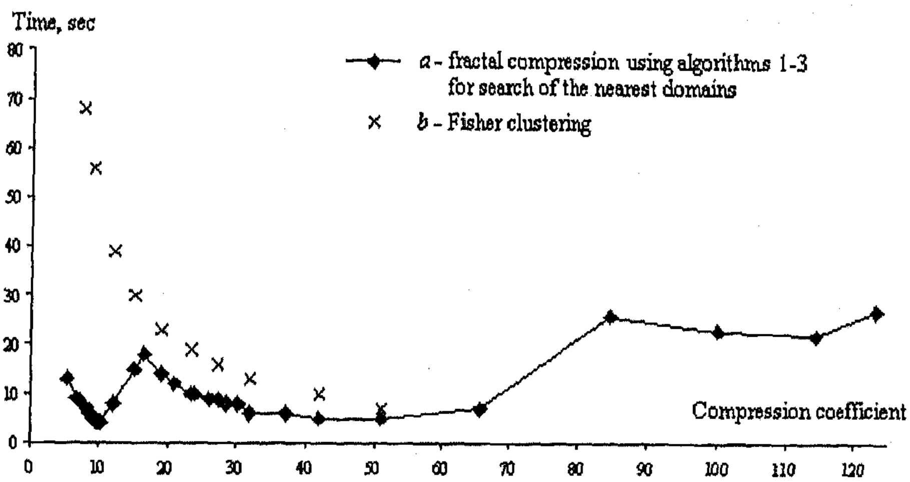

Figure 4

Conclusion and further developments

Though the high compression coefficients being achieved in the fractal method of compression of images, the main problem of this method is still the large computational complexity of the algorithm. In this work the effective method of optimization of the classical fractal algorithm is considered. This method speeds up considerably the search of optimal domain blocks, the sufficiently high compression coefficients being saved.

The degree of compression can be increased using the additional entropy coding methods [14]. It appears, that the use of elements of pattern recognition for more precise partition into clusters is also promising.

References

1. Lototskiy R.V., Application of transformation of wavelets for graphical images compression, *Problemy upravleniya i informatiki*, 2000, No. 4, 116–127.

2. Pennebaker W.B., Mitchell J.L., JPEG still image data compression standard, Van Nostrand Reinhold, N.Y., 1993.
3. Barnsley M.F., Fractals everywhere, Academic Press, San Diego, 1988.
4. Jacquin A.E., Image coding based on a fractal theory of iterated contractive image transformations, *IEEE Transactions on image processing*, 1992, **1**, No. 1, 18–30.
5. Ramamurthi B. and Gersho A., Classified vector quantization of images, *IEEE Transactions on Communication*, 1986, **34**, No. 11, 1105–1115.
6. Frigaard C., Gade J., Hemmingsen T., Sand T., Image compression based on a fractal theory, Internal report S701, Institute for electronic systems, aalborg university, Denmark, 1994.
7. Welstead S., Fractal and wavelet image compression techniques, SPIE optical engineering press, Washington, 1999.
8. Hamzaoui R., Codebook clastering by self-organizing map for fractal image compression, *Fractals*, World scientific publishing company, 1997, **57**, Supplementary issue, 27–28.
9. Ponomarenko N., Egiazarian K., Lukin V., Astola J., Speeding-up the fractal compression with clustering, *Proceedings from signal/image processing and pattern recognition*, Kyiv, 2000, 55–58.
10. Kohonen T., Self-organization maps, Springer series in information sciences, Springer-Verlag, 2001.
11. Bogdan A., Meadows H.E., Kohonen neural network for image coding based on iteration transformation theory, *Proceedings from SPIE neural and stochastic methods in image and signal processing*, 1992, **1766**, 425–436.
12. Kolmogorov A.N., Fomin S.V., The basics of the function theory and functional analysis [in Russian], Nauka, Moscow, 1976.
13. Kronover R.M., Fractals and chaos in dynamic systems: The foundations of the theory [in Russian], Postmarket, Moscow, 2000.
14. Bell T., Cleary J., Witten I., Text compression, Prentice Hall, Englewood Cliffs, New Jersey, 1990.

Authors of the Issue

Bodnya Alexey L.,
Institute of General Power Engineering of National Academy of Sciences of Ukraine, Kyiv

Deyev Anatoliy A.,
Institute of Information Recording Problems of National Academy of Sciences of Ukraine, Kyiv

Golovanov Igor N.,
Institute of General Power Engineering of National Academy of Sciences of Ukraine, Kyiv

Kachanov Pyetr T.,
Institute of Information Recording Problems of National Academy of Sciences of Ukraine, Kyiv

Kachanova Elena V.,
Institute of Information Recording Problems of National Academy of Sciences of Ukraine, Kyiv

Kirichenko Nikolay F.,
V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine, Kyiv

Korkhin Arnold S.,
Research Institute of Ferrous Metallurgy Automation, Dnepropetrovsk

Legeza Victor P.,
National Technical University of Ukraine "Kyiv Polytechnical Institute,.

Lototskiy Roman V.,
V.M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine, Kyiv

Matvienko Vladimir T.,
Kyiv National Taras Shevchenko University

Mizernyi Victor N.,
Vinnitsa State Technical University

Torba Lyubov T.,
Institute of Information Recording Problems of National Academy of Sciences of Ukraine, Kyiv

Totsenko Vitaliy G.,
Institute of Information Recording Problems of National Academy of Sciences of Ukraine, Kyiv

Tsyganok Vitaliy V.,
Institute of Information Recording Problems of National Academy of Sciences of Ukraine, Kyiv