JOURNAL OF ENGINEERING SCIENCES
ЖУРНАЛ ІНЖЕНЕРНИХ НАУК
ЖУРНАЛ ИНЖЕНЕРНЫХ НАУК

Web site: http://jes.sumdu.edu.ua

DOI: 10.21272/jes.2019.6(1).e3 Volume 6, Issue 1 (2019)

UDC 681.3.05

Increasing the Speed of Fractal Image Compression Using Two-Dimensional Approximating Transformations

Maydaniuk V. P., Arseniuk I. R., Lishchuk O. O.

Vinnytsia National Technical University, 95 Khmelnitsky Rd., 21021 Vinnytsia, Ukraine

Article info:

Paper received:	October 26, 2018
The final version of the paper received:	January 20, 2019
Paper accepted online:	January 29, 2019

***Corresponding Author's Address:**
igrosars@gmail.com

Abstract. Fractal image compression algorithm is known for allowing very high compression rates (the best examples – up to 1 000 times with acceptable visual quality) for real photos of natural objects, which is not possible for other lossy compression methods. The main disadvantage of the fractal method is the low rate of encoding, which is due to the fact that in order to obtain high image quality for each rank block, it is necessary to perform a search of all domain blocks, and for each domain block, at least eight affine transformations must be performed. Despite the large number of works devoted to increasing the speed of fractal images compression, it is worth noting that this problem remains very relevant. The aim of the work is to find methods for increasing the speed of fractal image compression. Based on the analysis of known approaches of increasing the fractal compression rate, a proposed method is based on the representation of rank and domain blocks in the form of coefficients of two-dimensional linear approximation, which allows for each rank block to perform a rapid pre-selection of blocks by three approximation coefficients. With the selected blocks, the transformations that are characteristic for fractal compression are performed. Since the quantity of the selected blocks is considerably less than the total number of domain blocks, one should expect a significant gain in the sealing speed. The simulation done in the Python programming language showed that the proposed method can increase the fractal image compression rate by on average of 10 times compared to Arnaud Jacquin's method without significant loss of image visual quality.

Keywords: image compression, fractal encoding, two-dimensional approximation, image fractal properties.

1 Introduction

Images that are presented in digital form must be stored on media and transmitted by communication channels. To save memory and make more efficient use of system resources, special encoding algorithms are creates [1, 6]. The image is a special kind of data that has redundancy in two dimensions, which provides additional opportunities for compression [1, 4]. One of the promising methods of image compression is a fractal method [1]. Fractal encoding is a mathematical process for encoding raster images that contain a real image in a set of mathematical data that describes the fractal properties of an image. This type of encoding is based on the fact that all natural and most artificial objects contain excessive information in the form of identical image blocks that are repeated. They got the name fractals. Fractal is a structure that consists of similar shapes and drawings that can be in different sizes.

2 Literature Review

The fractal compression algorithm is known for allowing very high densification factors (best examples – up to 1 000 times with acceptable visual quality) for real photographs of natural objects that can not be used for other lossy compression algorithms [2, 3].

The main disadvantage of the fractal method is the low rate of encoding, which is due to the fact that in order to obtain high image quality for each rank block, it is necessary to perform a search of all domain blocks, and for each domain block, at least eight affine transformations must be performed [7–9]. One of the possible efficient and fast image coding schemes by fractal method was proposed by Arnaud Jacquin [7]. But if you count the number of multiplication operations to find the coefficients of affine transformations of one rank block in the image in grayscale grays of 512×512 ($4^{4.5} = 512$, $k = 4.5$) pixels with the size of the rank block 4×4 ($n = 4$), the domain 8×8 and step of the choice of domain blocks 2, then even for the algorithm proposed by Jacquin, the total number of operations of multiplication will be quite large and will be [1]:

$$M = 8\cdot[4n^{k+1}(n^{k-1}-3)+9n^{2}] =$$
$$= 8\cdot[4\cdot4^{5.5}\cdot(4^{3.5}-3)+9\cdot4^{2}] = 8.2\cdot10^{6}.$$

Consequently, the purpose of increasing the rate of compression of images by the fractal method is very relevant. With improved performance, the fractal compression algorithm can become one of the most effective image compression algorithms [1].

3 Research Methodology

3.1 Mathematical model of encoding-decoding of images by fractal method

From a physical point of view, fractal encoding is based on the assertion that the image contains affinity redundancy. The mathematical model used in fractal image compression is called Iterated Function Systems (IFS). IFS contain a set of compression transformations that can be set for the image *S* as follows [1, 8]:

$$W(S) = w_i(S). \qquad (1)$$

According to the Banach theorem, there exists a certain class of mappings called pressing ones, and the following statement holds true for them: if, to some image f_0, we begin to repeatedly apply the mapping *W* in such a way that:

$$f_1 = W(f_0),\ f_i = W(f_{i-1}), \qquad (2)$$

then with *"i"* going to infinity we get the same image no matter what image we took for f_0:

$$f = \lim_{i\to\infty} f_i \qquad (3)$$

The image *f* is called a fixed conversion point *W* or attractor.

As transformations w_i affinity mapping is used:

$$w_i\begin{pmatrix}x\\y\\z\end{pmatrix} = \begin{bmatrix}a_i & b_i & 0\\ c_i & d_i & 0\\ 0 & 0 & S_i\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix} + \begin{bmatrix}dx\\dy\\O_i\end{bmatrix}, \qquad (4)$$

where a_i, b_i, c_i, d_i – affine coefficients of deformation, compression, rotation; d_x, d_y – coefficients of move; *x, y* – the coordinates of the point that is converted; *z* – its intensity. Parameter S_i controls contrast, and O_i – brightness of the image. Knowing the coefficients of these transformations, we can restore the original image.

The fractal image coding algorithm can be described as follows. The process of compression begins with the fact that the image is initially divided into non-overlapping (ranked areas), and then in the domain blocks that can overlap, as shown in Figure 1 [7].

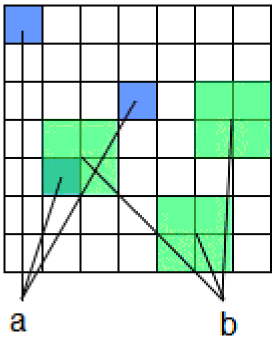


Figure 1 – Selection of blocks in the image:
a– rank, b – domain

Domains must have distinctive fragments that are then used to construct the decoded image. After that, the image encoding begins by selecting for each domain region of the most relevant domain, by which the brightness distribution in the ranked region can be approximated by the distribution of brightness in the domain. In order to get the best approximation, the domains are subjected to affine transformations, which result in not only their geometric deformation, but also changes in contrast and brightness. If the brightness distribution in the converted domain fails to achieve a satisfactory approximation of the brightness distribution in the rank region, the rank region is divided into four parts and the process is repeated. The quality of the required approximation is given in the form of an acceptable value of the average square of the approximation error (the mean square of the discrepancy). Domain numbers used in the encoding of each rank domain, as well as affinity overflow coefficients, are written to a file. The compressed image file contains a heading with information about the location of ranked domains and domains, as well as a table of effectively packed affine coefficients for each rank domain.

One of the possible patterns of image encoding by the fractal method, proposed by Arnaud Jacquin, contains the following steps [1, 8]:

– the image is divided into areas adjacent to each other by the size of *N*×*N* (ranked areas);

– a set of domain scopes is specified. Domain areas can overlap, they do not have to cover the entire surface of the image. The size of domain areas is *2N*×*2N*;

– the domain, which after affine transformations, most closely approximates the rank region is chosen for each rank. In practice, eight variants of mapping one square to another with usage of affine transformations are applied. These are turns of the image at angles 0°, 90°, 180°, and 270° relative to its center and symmetry transformation relative to the orthogonal axes, which pass through the center of the fragment, perpendicular to its sides.

The accuracy of the approximation *F* is determined by means of the mean square criterion:

$$F = \sum_{i,j}(Sd_{ij} + O_{ij} - r_{ij})^2, \qquad (5)$$

where d_{ij} – values obtained as a result of averaging fragments with dimensions of 2×2 of the domain's region elements, that leads it's size to the size of the rank region; r_{ij} – values of elements of the rank region. The displacement O_{ij} can be either a constant, or described by polynomials of the first, second, third order.

By equating partial derivatives of the expression *S* and *O* to zero:

$$\frac{\partial F}{\partial S}=0\,,\ \frac{\partial F}{\partial O}=0\,, \tag{6}$$

Let's find the values *S* and *O*, at which the minimum of the expression is reached:

$$O=\frac{1}{n^2}\cdot\left(\sum_{i,j}^{n}r_{ij}-S\sum_{i,j}^{n}d_{ij}\right), \tag{7}$$

$$S=\frac{n^2\sum\limits_{i,j}^{n}r_{ij}d_{ij}-\sum\limits_{i,j}^{n}r_{ij}\sum\limits_{i,j}^{n}d_{ij}}{n^2\sum\limits_{i,j}^{n}d_{ij^2}-(\sum\limits_{i,j}^{n}d_{ij})^2}. \tag{8}$$

Domain blocks are usually chosen with step *n*/2 at *n* = 4. The following parameters are written to the output file:

1) coordinates of the domain area with the lowest value of *F*;

2) values for *O* and *S*, obtained according to the formulas (7), (8);

3) number of affine transformation.

The decoding algorithm consists in the fact that two instances of the same image *A* and *B* are taken, the distribution of brightness in which is irrelevant. The areas whose boundaries coincide with boundaries of ranks and domains areas are selected on these images and then, using known values of affine coefficients, by the domains selected in image *B* brightness distributions in the rank areas of image A are found. After that, images *A* and *B* change places and the operation is repeated. It can be shown that with many repetition of this operation, the brightness distribution in images *A* and *B* will be closer to the brightness distribution in the original image. Let's pay attention to the fact that the algorithms of compression and decompression are asymmetric. It is also worth noting that the compression process takes much longer than the decompression process. Decoding of the compressed image is iterative and consists of the following steps:

1) two images of the same size *A* and *B* are created. The size of these images does not necessarily equal the size of the original image; the initial drawing of areas *A* and *B* does not matter;

2) the image *B* is divided into rank areas, as in the first stage of the compression process. For each rank area of image *B* an affine transformation of the corresponding domain area of image *A* is performed and the result is placed in *B*.

3) performed operations are identical to the previous item, only the images *A* and *B* swap places;

4) the second and third steps are repeated repeatedly until the images *A* and *B* do not become indistinguishable.

The main disadvantage of the fractal method is the low rate of encoding, which is due to the fact that in order to obtain high image quality for each rank block, it is necessary to go through all domain blocks and at least eight affine transformations must be performed for each domain block [5].

3.2 Known methods to increase the rate of fractal image compression algorithm

To improve the speed and efficiency of fractal image encoding, a number of optimization methods are used. The simplest and slowest way of fractal encoding is to check each domain block and perform calculations according to the expressions (5), (7), (8). This method is called an exhaustive search. When encoding images of natural origin, you can increase the coding speed by taking *S* = 1, since, taking into account the image statistics, there is always a domain block that approximates a given rank block with the required precision. Then from the expressions (5), (7) we get:

$$F=\sum_{i,j}(d_{ij}+O_{ij}-r_{ij})^2\,, \tag{9}$$

$$O=\frac{1}{n^2}\cdot\left(\sum_{i,j}^{n}r_{ij}-\sum_{i,j}^{n}d_{ij}\right). \tag{10}$$

The contrast of the decoded image can be restored by other methods. This simplification allows you to reduce the number of arithmetic operations by 60 % and, accordingly, increase the compression speed.

The most popular methods for increasing the speed of encoding images by fractal method are as follows [10]:

1) search for domain blocks that do not exceed the specified value;

2) local and sub-local search;

3) isometric prediction;

4) classification of domain and rank blocks, the ranked compares with domain blocks of the same class.

Among the methods should be noted the classification proposed by Arnaud Jacquin [1, 8]. It is based on the block topology and involves:

– blocks without contours;

– blocks, invariant to the orientation (texture blocks);

– contour blocks (exhaustive search).

3.3 Increasing the rate of the fractal image compression by two-dimensional approximation

To increase the speed of image compression under the Arnaud Jacquin scheme, it is proposed to perform a preliminary selection of domain blocks based on the approximation coefficients [7].

In the case of linear approximation, the pixel value for a two-dimensional image is determined as follows:

$$f(x,y)=ax+by+c. \tag{11}$$

In the general case, the values of *f*(*x*, *y*) differ from the value of the pixel *zxy*. The minimum distance value is achieved with a minimum value of the sum of squares of distances, that is:

$$S=\sum_{x=1}^{N}\sum_{y=1}^{M}(ax+by+c-z_{xy})^2=Min, \qquad (12)$$

where *M, N* – image size; z_{xy} – pixel value at the point of the image with the coordinates *x, y*.

The function *S* has a minimal extremum at the point where partial derivatives of the coefficients are zero:

$$\frac{\partial S}{\partial a}=0,\ \frac{\partial S}{\partial b}=0,\ \frac{\partial S}{\partial c}=0. \qquad (13)$$

Thus, we obtain a system of three equations for three unknowns. For ranked blocks with a size *n* = 4, the system of equations is as follows:

$$\begin{cases} 120a+100b+40c=\sum_{y=1}^{4}\sum_{x=1}^{4}z_{xy}x \\ 100a+120b+40c=\sum_{y=1}^{4}\sum_{x=1}^{4}z_{xy}y \\ 40a+40b+16c=\sum_{y=1}^{4}\sum_{x=1}^{4}z_{xy} \end{cases} \qquad (14)$$

Having solved the system of equations (14), it is possible for each rank and domain blocks to determine the coefficients of approximation *a, b, c*.

Therefore, the encoding process will have the following additional steps:

1. Each domain and rank block is presented in the form of coefficients of approximation. For *n* = 4, the approximation coefficients from (14) are calculated as follows:

$$b=\frac{-3\sum_{x=1}^{4}\sum_{y=1}^{4}z_{xy}+1{,}2\sum_{x=1}^{4}\sum_{y=1}^{4}z_{xy}y}{24}; \qquad (15)$$

$$c=\frac{3\sum_{x=1}^{4}\sum_{y=1}^{4}z_{xy}-\sum_{x=1}^{4}\sum_{y=1}^{4}z_{xy}x-20b}{8}; \qquad (16)$$

$$a=\frac{3\sum_{x=1}^{4}\sum_{y=1}^{4}z_{xy}-\sum_{x=1}^{4}\sum_{y=1}^{4}z_{xy}y-8c}{20}. \qquad (17)$$

2. For each rank block, pre-selection of domain blocks is performed by three coefficients of approximation, for example, by quadratic deviation:

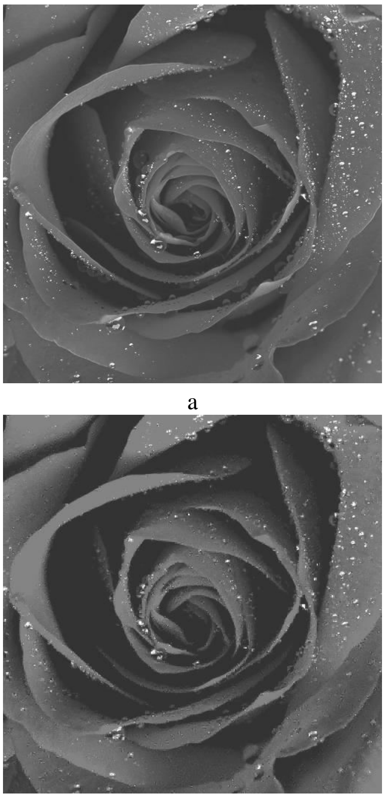
$$\begin{aligned} S_{rd1}&=(a_r-a_d)^2+(b_r-b_d)^2+(c_r-c_d)^2; \\ S_{rd2}&=(a_r-b_d)^2+(b_r-a_d)^2+(c_r-c_d)^2, \end{aligned} \qquad (18)$$

where a_r, b_r, c_r – coefficients of approximation for the rank block; a_d, b_d, c_d – coefficients of approximation for the domain block.

With the selected blocks, the transformations characteristic of fractal compaction by the Jacquin method are performed. Since the selected blocks number is considerably less than the total number of domain blocks, one should expect a significant gain in speed.

4 Results

The simulation executed in the Python programming language showed that the proposed method for increasing the speed of fractal image compression can achieve acceleration of 5–10 times compared to the Arnaud Jacquin's method (exhaustive search) without serious loss of image visual quality (Figure 2).



a

b

Figure 2 – Results of simulation of high-speed fractal compression: a – original image; b – image restored after encoding by the proposed method

For example, on the same computer, it takes about 52 minutes to encode an image of 512×512 by the method proposed by Arno Jacquin, and for encoding the image according to the method proposed above – only 5 min, that is, the encoding speed increased by 10 times.

To compare the proposed method and Arnaud Jacquin's method in Table 1, results are presented for images of different sizes. The larger size of the image, the better results are provided by the proposed method, since the pre-selection reduces the space for finding domain blocks for each rank block.

Table 1 – Comparison of the encoding time of images of various sizes

Image size	exhaustive search tg, s	proposed method ts, s	Acceleration rate, tg/ts
128×128	9.7	1.4	6.9
256×256	196	24	8.3
512×512	3 125	302	10.3
1024×1024	19 333	1 588	12.2

5 Conclusions

The main disadvantage of the fractal method is the low rate of encoding, which is due to the fact that in order to obtain high image quality for each rank block, it is necessary to go through all domain blocks and at least eight affine transformations must be performed to each domain block. The method for increasing the fractal compression speed by proposing rank and domain blocks in the form of coefficients of approximation is proposed. This allows to perform a quick pre-selection of domain blocks, which ultimately increases the fractal seal speed on average by 10 times.

REFERENCES

1. Welstead, S. (1999). *Fractal and wavelet image compression techniques*. Bellingham, SPIE Optical, Engineering Press.
2. Fisher, Y., Jacobs, E. W., & Boss, R. D. (1991). *Fractal image compression using iterated transform*.
3. Jacquin, A. (1992). Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, pp. 18–30.
4. Ghosh, S. K., Mukherjee, J., & Das, P. P. (2004). Fractal image compression: a randomized approach. *Pattern Recognition Letters*, Vol. 25, pp. 1013–1024.
5. Thomas, L., & Deravi, F. (1995). Region-based fractal image compression using heuristic search. *IEEE Transactions on Image Processing*, Vol 4, pp. 832–838.
6. Maydanyuk, V. P. (2001). *Metody i zasoby komp'yuternykh informatsiynykh tekhnolohiy. Koduvannya zobrazhen'*. Vinnytsya: VDTU [in Ukrainian].
7. Kozhem'yako, V. P., Maydanyuk, V. P., Zhukov, K. M., Khamdi, R. R., & Pika, S. O. (1999). Fraktal'ne stysnennya zobrazhen' pryrodnoho pokhodzhennya. *Vymiryuval'na ta Obchyslyuval'na Tekhnika v Tekhnolohichnykh Protsesakh*, Vol. 2, pp. 50–54.
8. Kozhemiako, V. P., Maidanuk, V. P., Pika, S., & Zhukov, K. M. (2001). Speeding up of fractal image compression. *Proceeding of SPIE*, Vol. 4425, pp. 9–16.
9. Maydanyuk, V. P., Lishchuk, O. O., & Korol', D. C. (2017). Aspekty optymizatsiyi shvydkosti fraktal'noho ushchil'nennya zobrazhen'. *Optyko-Elektronni Informatsiyno-Enerhetychni Tekhnolohiyi*. Vol. 1(33), pp. 24–32 [in Ukrainian].
10. Maydanyuk, V. P., & Lishchuk, O. O. (2018). Pidvyshchennya shvydkosti fraktal'noho metodu ushchil'nennya zobrazhen' za rakhunok aproksymatsiyi. *Internet – Education – Science*, pp. 7–9 [in Ukrainian].

Підвищення швидкості фрактального ущільнення зображень з використанням двовимірних апроксимуючих перетворень

Майданюк В. П., Арсенюк І. Р., Ліщук О. О.

Вінницький національний технічний університет, вул. Хмельницьке Шосе, 95, 21021, м. Вінниця, Україна

Анотація. Алгоритм фрактального ущільнення зображень відомий тим, що у деяких випадках дозволяє отримати дуже високі коефіцієнти ущільнення (найкращі приклади – до 1000 разів за прийнятної візуальної якості) для реальних фотографій природних об'єктів, що є неможливим для інших алгоритмів ущільнення зображень із втратами. Основним недоліком фрактального методу є низька швидкість кодування, яка пов'язана з тим, що для отримання високої якості зображення для кожного рангового блоку необхідно виконати перебір усіх доменних блоків, і для кожного доменного блоку необхідно виконати не менше восьми афінних перетворень. Незважаючи на велику кількість праць, присвячених підвищенню швидкості фрактального ущільнення зображень, варто констатувати, що дана проблема залишається актуальною. Метою роботи є пошук методів підвищення швидкості фрактального ущільнення зображень. На основі аналізу відомих підходів підвищення швидкості фрактального ущільнення запропоновано метод, який ґрунтується на поданні рангових та доменних блоків у вигляді коефіцієнтів двовимірної лінійної апроксимації, що дозволяє для кожного рангового блоку виконати швидкий попередній відбір доменних блоків за трьома коефіцієнтами апроксимації. З відібраними блоками виконуються перетворення, характерні для фрактального ущільнення. Оскільки обраних блоків значно менше загальної кількості доменних блоків, то слід очікувати значного збільшення швидкості ущільнення. Моделювання, виконане із застосуванням мови програмування Python, показало, що запропонований метод дозволяє підвищити швидкість фрактального ущільнення зображень у середньому в 10 разів порівняно з методом за схемою А. Жакена без суттєвих втрат візуальної якості зображення.

Ключові слова: ущільнення зображень, фрактальне кодування, двовимірна апроксимація, фрактальні властивості зображення.