

## On the Application of Invariant Image Parameters to Fractal Encoding of Bi-Level Images

Jonas VALANTINAS, Tomas ŽUMBAKIS

*Department of Applied Mathematics, Kaunas University of Technology  
Studentų 50, LT-51368 Kaunas, Lithuania  
e-mail: jonas.valantinas@ktu.lt*

Received: July 2006

**Abstract.** Fractal image compression is an engaging and worthwhile technology that may be successfully applied to still image coding, especially at high compression ratios. Unfortunately, the large amount of computation needed for the image compression (encoding) stage is a major obstacle that needs to be overcome. In spite of numerous and many-sided attempts to accelerate fractal image compression times, the “speed problem” is far from being carried to its conclusion.

In the paper, a new version (strategy) of the fractal image encoding technique, adapted to process bi-level (black and white) images, is presented. The strategy employs the necessary image similarity condition based on the use of invariant image parameters (image smoothness indices, image coloration ratios, etc.). It is shown that no images can be similar (in the mean squared error sense) if their respective parameter values differ more than somewhat. In the strategy proposed, the necessary image similarity condition plays a key role – it is applied to speed-up the search process for optimal pairings (range block-domain block), i.e., it enables to narrow the domain pool (search region) for each range block. Experimental analysis results show that implementation of the new fractal image encoding strategy accelerates bi-level image compression times considerably. Exceptionally good results (compression times and quality of restored images) are obtained for silhouette images.

**Key words:** digital images, invariant image parameters, fractal image compression.

### 1. Introduction

Throughout the last two short decades development and implementation of fractal image processing technologies was an area of increasing interest. Much was done in preparing and maturing diversified fractal image encoding schemes (Jacquin, 1992; Fisher, 1994; Saupe, 1995; Wohlberg and Jager, 1999; Tong and Wong, 2002; Furoo and Hasegawa, 2004; Žumbakis and Valantinas, 2005). Constant attention was paid to the development of fractal modelling techniques, to the analysis and synthesis of fractal images (Peitgen *et al.*, 1992; Turner *et al.*, 1998; Kaandorp and Kuebler, 2001; Valantinas and Žumbakis, 2004). Elements and thoughts of fractal geometry (fractal dimension, fractal interpolation, fractional integrals and derivatives) were put to work in real-world applications (Yokoya *et al.*, 1989; Samko *et al.*, 1993; Evans, 1997; Dimri, 2000; Blackledge *et al.*, 2004).

In the field of digital images (real-world image models) the fractal approach is of outmost importance, because it facilitates interpretation of the information content of an image. To say more, it provides us with a powerful tool to catch sight of a fundamental real-world image property generally known as self-similarity. Due to this property, the research and development of algorithms (“fractal techniques”) to extract important fractal characteristics from appropriate digital data has received significant attention in recent years. One of the most rapidly developing areas is the use of fractal geometry for performing image data compression.

Those, who are gone deep into the essence of the matter, comprehend that merely the extractability of self-similarity, found within images, made it possible to construct the fractal representation of an image. A. Jacquin was the first to propose a practical block based fractal image coding scheme (idea) in 1992, basis of most published fractal image coding schemes. A detailed introductory review on fractal image coding techniques can be found in (Fisher, 1994; Saupe and Hamzaoui, 1994).

However, the encoding complexity of the fractal image coding is extremely high and has become the major obstacle for its workable practical applications. The most computationally intensive part of the fractal encoding process is the searching step. If a brute force approach (exhaustive search) to the detection of optimal pairings “range block–domain block” is used, the fractal encoding complexity is always dominated by this searching process. In the past ten years, many interesting and promising strategies (searching algorithms) have been developed to get over the heavy computation of exhaustive search. Block categorization is the most commonly used method in speeding up the searching process (Bani-Eqbal, 1995; Hamzaoui, 1997; Wohlberg and Jager, 1999; Hamzaoui and Saupe, 2000). The main idea of block categorization is to sort out the fixed size range and domain blocks into different classes according to the image features (mean, variance, edge types, etc.). The searching process is then restricted to the same category (class), i.e., only those domain blocks which belong to the same class as the encoded range block will be matched, and the others are not considered. The computational complexity is reduced because fewer domain blocks are compared with the given range block. Another idea to accelerate the searching process at the encoding stage is bound up with a helpful observation that if the search is not totally exhaustive the resulting pairings may not be optimal but could be very close, with the result that only a minor loss of quality will occur. The simplest implementation of the idea goes under the name – “the nearest neighbor search strategy” (Caso *et al.*, 1996; Polvere and Napi, 2000; Tong and Wong, 2002; Saupe, 2002). The essence of the latter – only those domain blocks which partially overlap the encoded range block to some extent (or, lie at a certain distance away) are taken into consideration and analysed.

The design of efficient domain search strategies (searching algorithms) has consequently been one of the most active areas of research in fractal coding, resulting in a wide variety of solutions (Truong *et al.*, 2000; Tong and Pi, 2001; Furaio and Hasegawa, 2004; Hassaballah *et al.*, 2005). In parallels, we must admit that, despite all enumerated attempts to accelerate fractal image compression times, the “speed” problem is far from being carried to its conclusion.

Not long ago, we have developed a new strategy for the fractal encoding of grey-level images (Valantinas *et al.*, 2002; Žumbakis and Valantinas, 2005). The range blocks and corresponding domain blocks were categorized depending on their smoothness parameter values, introduced, from the first, to characterize manifestation of high frequency components in the image. The searching of the best matched domain block was carried out between the neighboring (or, within the same) smoothness classes. The computational complexity of the fractal image encoding process has been reduced considerably.

In the paper, a new version of the developed strategy (Žumbakis and Valantinas, 2005), this time oriented to process bi-level (black and white) images, is presented and analysed. The version is based, mainly, on the use of reasonably defined invariant image parameters (image smoothness indices, image coloration ratios, etc.). These parameters are employed to state the necessary image similarity condition, which is used later on to achieve image compression speed gains in the search for optimal pairings “range block-domain block”. Theoretical and experimental analysis results show that the developed bi-level image coding technique is rather efficient in solving the earlier mentioned “speed” problem, i.e., in achieving fractal image compression speed gains. Exceptionally good results (image compression times and quality of restored images) are obtained for silhouette images.

The rest of this paper is organized as follows. Section 2 introduces some invariant image parameters, associated with bi-level images, describes procedures for finding parameter values, formulates their properties and interprets possible application areas. Section 3 reviews the basic scheme of fractal image coding (Jacquin’s idea) and presents a new approach to achieving fractal (bi-level) image compression speed gains. In the centre of attention – application of the necessary image similarity condition to the development of an efficient search technique. Experimental results and some reconstructed images are shown in Section 4.

## 2. Invariant Image Parameters – Definition, Evaluation, Properties

Consider a set of digital images  $S^2(n) = \{[X(m)] \mid m = (m_1, m_2) \in I^2\}$ , where  $I = \{0, 1, \dots, N - 1\}$ ,  $N = 2^n$ ,  $n \in \mathbb{N}$ ;  $X(m) \in \{0, 1, \dots, 2^p - 1\}$ , for all  $m \in I^2$ ;  $p$  ( $p \geq 1$ ) equals the number of bits per pixel in the image (for bi-level images,  $p = 1$ ).

The distance (mean squared error)  $\delta$  between any two elements of the set  $S^2(n)$  – images  $[X_1(m)]$  and  $[X_2(m)]$  – is specified by

$$\delta = \delta(X_1, X_2) = \left( \frac{1}{N^2} \sum_{m \in I^2} (X_2(m) - X_1(m))^2 \right)^{1/2}. \quad (1)$$

Let us denote the two-dimensional discrete spectrum (Walsh-Hadamard (WHT), cosine (DCT), etc. (Ahmed and Rao, 1975)) of the image  $[X(m)] \in S^2(n)$  by  $[Y_X(k)]$ . It is well known that the spectral coefficients  $Y_X(k)$  ( $k = (k_1, k_2) \in I^2$ ) decrease in absolute value, as their serial numbers  $k$  (indices  $k_1$  and  $k_2$ ) increase, provided the basis vectors of

the discrete transform in use are presented in a frequency order. The latter circumstance implies that there exists a hyperbolic surface

$$z = z(x_1, x_2) = C/(x_1 \cdot x_2)^\alpha \quad (C \geq 0, \alpha \geq 0), \quad (2)$$

which approximates the ordered array of spectral coefficients  $\{|Y_X(k)| \mid k = (k_1, k_2) \in I^2, k_1^2 + k_2^2 \neq 0\}$  in the mean squared error sense, i.e.,

$$\left( \frac{1}{N^2 - 1} \sum_{\substack{k \in I^2 \\ (k_1^2 + k_2^2 \neq 0)}} \left( |Y(k)| - \frac{C}{(\bar{k}_1 \cdot \bar{k}_2)^\alpha} \right)^2 \right)^{1/2} \rightarrow \min; \quad (3)$$

here  $\bar{k}_i = \max\{k_i, 1\}$ ,  $i = 1, 2$ .

The quantity  $\alpha$  (expression (2)), characterizing the shape of the hyperbolic surface, i.e., the rate of decay of spectral coefficients, as their serial numbers increase, is assumed, in what follows, to be the smoothness parameter (class) of the image  $[X(m)] \in S^2(n)$ . This assumption is intuitively understandable – the more intense manifestation of high frequency components in the discrete spectrum of the image, the more noticeable changes of pixel intensity values (sharp edges) are detected in the image.

Below, we present a means solely for finding the very first approximation of the image smoothness parameter value  $\alpha$ . Let us designate the set of indices of nonzero spectral coefficients, in the discrete spectrum  $[Y_X(k)]$  of the image  $[X(m)] \in S^2(n)$ , as  $H$ , i.e.,

$$H = \{k = (k_1, k_2) \in I^2 \mid Y_X(k) \neq 0, k_1^2 + k_2^2 \neq 0\}. \quad (4)$$

Then, application of the least squares method leads to the following result

$$\alpha = \alpha_0 = \frac{1}{A_N} \sum_{k \in H} (B_N - |H| \cdot P(k)) \cdot \log |Y_X(k)|, \quad (5)$$

where  $A_N = |H| \cdot C_N - B_N^2$ ;  $B_N = \sum_{k \in H} P(k)$ ;  $C_N = \sum_{k \in H} P^2(k)$ ;  $P(k) = \log(\bar{k}_1 \cdot \bar{k}_2)$ , for all  $k \in H$ ; by the way,  $A_N = 0$  if and only if the set  $H$  is empty, i.e., the digital image  $[X(m)]$  is absolutely smooth. It is worth emphasizing that the above “rough” image smoothness estimate (expression (5)) sometimes serves the purpose, especially, when one is interested in the difference between the smoothness classes of particular digital images.

To make the estimate more precise, various approaches are applied, namely: successive coordinate optimization procedures, the least squares method, special iterative techniques, etc. (Valantinas and Žumbakis, 2004). Experimental results show that the real world image smoothness parameter values, obtained using DCT or WHT, fall into the interval  $[0; 3)$ .

The smoothness class of bi-level (black and white) images can be defined in a new way as well. Here, we present one criterion for smoothness analysis of black and white images.

Let  $[X(m)] \in S^2(n)$ ;  $X(m) = X(m_1, m_2) \in \{0, 1\}$ , for all  $m_1, m_2 = 0, 1, \dots, N - 1$ ;  $n = \log_2 N$ . The quantity (criterion)

$$\alpha = \alpha_X = 1 - \left( \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-2} X(m_1, m_2) \oplus X(m_1, m_2 + 1) + \sum_{m_2=0}^{N-1} \sum_{m_1=0}^{N-2} X(m_1, m_2) \oplus X(m_1 + 1, m_2) \right) \frac{1}{2N(N - 1)}, \quad (6)$$

that expresses the total number of changes (in pixel values) along the spatial axes of the image, is assumed to be the smoothness index (parameter, class) of the bi-level image  $[X(m)]$ ; here “ $\oplus$ ” signifies the logical operation “EXCLUSIVE-OR”. It can be easily seen that the introduced image smoothness index (expression (6)) takes values from the interval  $[0, 1]$ .

Image smoothness parameters  $\alpha$  (expressions (5) and (6)) possess two exceptionally important (from the standpoint of practical applications) properties:

- invariance with respect to the isometric transformations (rotation, reflection, inversion, etc.), acting upon the image  $[X(m)]$ , i.e., if  $[\hat{X}(m)] \in S^2(n)$  represents a transformed version of the initial image  $[X(m)]$ , then smoothness parameter values of  $[\hat{X}(m)]$  and  $[X(m)]$  coincide, i.e.,  $\alpha_{\hat{X}} = \alpha_X$ ;
- continuity of  $\alpha: S^2(n) \rightarrow \mathbb{R}$ , grasped in the way that small changes in the image  $[X(m)] \in S^2(n)$  call forth small changes in the smoothness parameter value  $\alpha$ , i.e., if  $[\tilde{X}(m)] = [X(m) + \Delta X(m)] \in S^2(n)$  and  $\delta(X, \tilde{X}) \leq \delta_0$ , then  $|\alpha_X - \alpha_{\tilde{X}}| \leq \varepsilon_0$ ; here  $\varepsilon_0$  and  $\delta_0$  are small positive numbers.

The detailed proof of the above properties (for the case of grey-level images (expression (5)), as well as explanatory description of usable isometric transformations, can be found in (Valantinas and Žumbakis, 2004). In the case of bi-level images, the invariance property of the image smoothness index  $\alpha$  (expression (6)) follows directly from its definition. Continuity of  $\alpha$  is ensured by a simple observation – inversion of  $k$  ( $1 \leq k < N^2$ ) pixel values in the (bi-level) image  $[X(m)]$  results in an increment  $\Delta\alpha$  of the smoothness index  $\alpha$ , that never exceeds  $2k/N(N - 1)$ .

On the basis of continuity property of the smoothness parameter (index)  $\alpha$ , we can immediately state the necessary image similarity condition

$$(\delta(X_1, X_2) \leq \delta_0) \Rightarrow (|\alpha_{X_1} - \alpha_{X_2}| \leq \varepsilon_0) \quad (7)$$

(here  $\alpha_{X_1}$  and  $\alpha_{X_2}$  are smoothness parameter (index) values for images  $[X_1(m)] \in S^2(n)$  and  $[X_2(m)] \in S^2(n)$ , respectively), which asserts that two particular images cannot be similar if their smoothness classes (smoothness parameter values) differ more than somewhat. The relationship between  $\varepsilon_0$  and  $\delta_0$ , obviously, should be established experimentally.

For stating of the necessary image similarity condition, the use can also be made of the invariant (bi-level) image parameter  $\nu$ , specified by the following formula:

$$\nu = \nu_X = \left| 1 - \frac{2}{N^2} \sum_{m_2=0}^{N-1} \sum_{m_1=0}^{N-1} X(m_1, m_2) \right|. \quad (8)$$

The introduced parameter  $\nu = \nu_X$  expresses the relative number of “ones” (or “zeros”) in the image  $[X(m)]$  and is called the image coloration ratio.

From (8) it directly follows that the image coloration ratio  $\nu$  continuously depends on the changes in the image  $[X(m)]$ . Moreover, the ratio  $\nu$  is invariant with respect to the isometric transformations (rotation, reflection, inversion, etc.), acting upon the image.

Despite the fact that  $\nu$  cannot be used in the role of the image smoothness parameter value, it gives us a chance to formulate anew the necessary image similarity condition. Really, for two bi-level images  $[X_1(m)] \in S^2(n)$  and  $[X_2(m)] \in S^2(n)$ , we have

$$(\delta(X_1, X_2) \leq \delta_0) \Rightarrow (|\nu_{X_1} - \nu_{X_2}| \leq \varepsilon_0) \quad (9)$$

(here  $\nu_{X_1}$  and  $\nu_{X_2}$  are coloration ratio values for images  $[X_1(m)]$  and  $[X_2(m)]$ , respectively).

Now, a truly valuable conclusion can be made – since the transformed (rotated, reflected, etc.) bi-level images preserve smoothness parameter values, the necessary image similarity condition (expression (7) or (9)), based on the application of invariant image parameter values, can be efficiently employed to accelerate the searching process in block based fractal image coding procedures, i.e., to narrow the search region for the best pairings “range block – domain block”.

### 3. Accelerating Fractal Compression Times for Bi-Level Images

The fractal image encoding (compression) principle consists in finding a construction rule that produces a fractal image which approximates the original image. Fractal image compression is based on the mathematical theory of iterated function systems (IFS), developed by M. Barnsley (1993). A. Jacquin was the first to propose a practical block based fractal image compression scheme (idea, strategy) based on the so-called partitioned iterated function systems (PIFS) (Jacquin, 1992; Barnsley, 1993).

To introduce the relevant terminology and notations, a brief explanation of the Jacquin’s basic idea is presented below.

#### 3.1. The Block Based Fractal Image Encoding Idea

In its simplest form (Jacquin’s approach), the image to be processed  $[X(m)] \in S^2(n)$  is partitioned at two scales (one twice the other), i.e., into the fixed size range blocks  $[U(m)] \in S_1^2(3) \subset S^2(3)$  and domain blocks  $[V(m)] \in S_1^2(4) \subset S^2(4)$ . The former (range) blocks are non-overlapping and contain every pixel. The latter ones (domain blocks) may overlap and not necessarily contain every pixel. The essence of the

approach is the pairing of each range block  $[U(m)]$  to a domain block  $[V(m)]$  such that  $\delta = \delta(U, V)$  is minimal.

The search for the best domain block for a particular range block is complicated by the requirement that the range block, in general, matches the transformed version of a domain block. In other words, the problem is to find for each range block the domain block that can be made closest (in the sense of  $\delta$ ) by an admissible isometric transformation (rotation, reflection, etc.). Without any doubt, computation required is enormous (“speed problem”;  $\sim 0.5 \cdot 10^9$  pairings, for images  $256 \times 256$ ;  $\sim 8.1 \cdot 10^9$  pairings, for images  $512 \times 512$ ).

To accelerate the pairing process in the encoding stage, a helpful observation is that if the search is not exhaustive the resulting PIFS may not be optimal but could be close enough, and a minor loss of quality will likely occur at the image decompression stage.

As it was mentioned above, the searching process may be simplified by introducing an appropriate invariant representation for each image block. Transforming range and domain blocks to this representation allows direct distance comparisons between them to find out the best possible match.

We here present a new development (approach, strategy) in overcoming the “speed problem” mentioned above. The proposed approach is oriented to process bi-level (black and white) images, employs regular segmentation of the image into non-overlapping range blocks and applies the necessary image similarity condition, based on the direct use of invariant image parameter values (Section 2). It is shown that image compression time savings are obtained irrespectively of the size of range (domain) blocks the image under processing is partitioned into.

### 3.2. *Implementation of the Necessary Image Similarity Condition*

In most cases, the similarity between any two images (or, between any two fragments (blocks) of the same image) is confirmed (or, denied) using the earlier introduced metrics  $\delta$  (expression (1); Section 2), i.e., two bi-level images,  $[U(m)] \in S^2(n)$  and  $[V(m)] \in S^2(n)$ , are considered to be similar, if and only if  $\delta = \delta(U, V) \leq \delta_0$  ( $\delta_0$  being a priori fixed small positive number); otherwise ( $\delta(U, V) > \delta_0$ ), images  $[U(m)]$  and  $[V(m)]$  are said to be dissimilar.

The necessary image similarity condition follows directly from the continuity property stated above (Section 2), i.e.,

$$(\delta(U, V) \leq \delta_0) \Rightarrow (|\theta_U - \theta_V| \leq \varepsilon_0), \tag{10}$$

where  $\theta_U$  and  $\theta_V$  signify the invariant parameter values – smoothness indices  $\alpha$  (expression (7); Section 2), or coloration ratios  $\nu$  (expression (9); Section 2) – for images  $[U(m)]$  and  $[V(m)]$ , respectively. In other words, two bi-level images (blocks) can not be similar if their respective parameter values differ more than somewhat ( $|\theta_U - \theta_V| > \varepsilon_0$ ).

Particularly important detail – the invariance property of image smoothness estimates (Section 2) makes the above necessary condition adaptable with the earlier mentioned

transformations (rotation, reflection, etc.), acting upon the image (domain block)  $[V(m)]$  (Section 3.1).

The developed fractal (bi-level) image encoding strategy employs the necessary image similarity condition (expression (10)), which makes it possible to reduce the domain pool for each range block to a considerable extent. At the same time, the best pairings “range block – domain block” are not missed (Fig. 1).

The main points of the developed strategy are presented below.

Firstly, for finding numerical values of the invariant image parameters in use, each range block  $[U_i(m)] \in S_1^2(n_1) \subset S^2(n_1)$ , as well as each domain block  $[V_j(m)] \in S_1^2(n_2) \subset S^2(n_2)$ , in the bi-level image  $[X(m)] \in S^2(n)$  is looked over only once (here  $i = 1, 2, \dots, 4^{n-n_1}$ ;  $j = 1, 2, \dots, (2^n - 2^{n_2} + 1)^2$ ; usually,  $n_1 \in \{2, 3\}$  and  $n_2 = n_1 + 1$ ). As a result, two sequences of invariant image parameter values,  $\{\theta_{U_i}\}$  and  $\{\theta_{\hat{V}_j}\}$ , are formed ( $[\hat{V}_j(m)] \in S^2(n_1)$  is a shrunken copy of the domain block  $[V_j(m)]$ ). So, domain and range blocks are categorized into a finite number of classes, according to their invariant representations – image smoothness index values (or, image coloration ratio values). The latter circumstance makes unnecessary repeated evaluation of values of  $\delta = \delta(U, \hat{V})$  in the searching process of optimal pairings “range block-domain block”. Each range block, now, is immediately associated with a set (pool) of candidate domain blocks, by comparing numerical values of their invariant parameters, i.e., all the blocks falling into the domain pool, associated with a particular range block, satisfy the necessary image similarity condition (expression (10)).

Secondly, a simple analysis of algebraic expressions used in finding numerical values of the invariant parameter  $\theta$  for all bi-level image blocks, in obtaining shrunken copies of domain blocks, as well as in establishing the fact of block similarity, shows that the time expenditures, associated with these steps, are equal to (suppose:  $n_1 = 3$ ;  $n_2 = 4$ ;  $\tau_a$  and  $\tau_m$  denote the time expenditure required to perform a single addition and multiplication operations, respectively;  $\theta = \nu$ ):

- (1)  $256 \cdot \tau_a + 64 \cdot \tau_m$  (for obtaining the shrunken copy  $[\hat{V}(m)]$  of  $[V(m)]$ );
- (2)  $65 \cdot \tau_a + 2 \cdot \tau_m$  (for computing coloration ratio value  $\nu$  of  $[U(m)]$  (or  $[\hat{V}(m)]$ );
- (3)  $1024 \cdot \tau_a + 2 \cdot \tau_m$  (for establishing image similarity fact between two blocks,  $[U(m)]$  and  $[\hat{V}(m)]$ ; isometric transformations inclusive).

Combining these results, under assumption that  $\tau_a = \tau_m$ , the following expression for the total image compression time expenditure  $\tau_{total}$  is obtained:

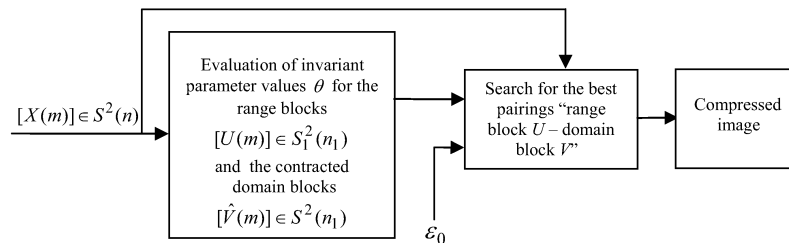


Fig. 1. Fractal bi-level image encoding scheme – implementation of the necessary image similarity condition.



$$\tau_{total} = \left( \frac{67}{64}N^2 + 387(N - 15)^2 + \frac{513}{328}N^2M \right) \cdot \tau_a; \tag{11}$$

here  $M$  stands for the averaged number of domain blocks  $[V(m)] \in S_1^2(4) \subset S^2(4)$ , contained in the domain pool (search region) of the range block  $[U(m)] \in S_1^2(3) \subset S^2(3)$ ;  $N \times N$  is the size of the initial image  $[X(m)] \in S^2(n)$ .

On the other hand, the total time expenditure ( $\tau_{total}^\circ$ ), associated with the exhaustive search for optimal pairings “range block–domain block” (Jacquin’s approach), equals:

$$\tau_{total}^\circ = \left( 320(N - 15)^2 + \frac{513}{64} \cdot N^2(N - 15)^2 \right) \cdot \tau_a. \tag{12}$$

The comparative analysis of expressions (11) and (12) gave preprocessing results for the prospective fractal image compression speed gains, i.e., for the acceleration of image encoding times – for the ratio  $k = \tau_{total}^\circ/\tau_{total}$  (Table 1). As it can be seen, even for  $N = 256$ , the ratio  $k = \tau_{total}^\circ/\tau_{total}$  exceeds 200, provided  $M \leq 10^3$ ; for  $M \leq 10^4$ , “success” (speed gains) is ensured unconditionally.

Similar reasoning can be applied to derive quantitative time expenditure results for other cases (including new sizes of the range (domain) blocks and other invariant image parameters).

It was found out that for black and white images, additional image compression time gains are achievable. They have links with slight modifications of the proposed fractal image coding strategy (Fig. 1). The main change – only non-monochrome range blocks of the binary image are processed at the encoding stage, whereas monochrome range blocks escape analysis, i.e., they are marked by the value 0 (black block) or 1 (white block) at once. One can easily ascertain that such a modification (in many cases) leads to noticeable improvements of the bi-level image compression effect. Suppose, the size of the image under processing  $[X(m)]$  is  $256 \times 256$ , and the size of the range block  $U$  is  $8 \times 8$ . If the format of the compressed data (describing the set of best pairings “ $U - V$ ”) is chosen to be  $\langle e, f, t, o \rangle$  (here  $e$  and  $f$  specify coordinates of the domain block  $V$  for a current range block  $U$  ( $8 \times 2 = 16$  bits needed),  $t$  specifies the type of the isometric transformation (3 bits) and  $o$  characterizes the type of the range block  $U$  (2 bits needed)), then the total number of bits attached to encode the given image  $[X(m)]$  equals

Table 1  
Fractal image compression speed gains (in the sense of  $k = \tau_{total}^\circ/\tau_{total}$ ;  $\theta = \nu$ )

The averaged number of blocks in a domain pool, $M$	Image size, $N \times N$		
	128×128	256×256	512×512
$10^2$	223.50	930.88	3792.78
$10^3$	54.96	244.14	1026.17
$10^4$	6.44	29.14	123.72
$10^5$	–	2.97	12.63

$21 \cdot (256/8) \cdot (256/8) = 21504$ . So, the image compression ratio equals  $65536/21504 \cong 3.05$ . The result, evidently, doesn't meet any requirements. Is it possible to make the image compression effect better? The answer is positive, provided the bi-level image under processing is of silhouette type. In the latter case, the bi-level image contains quite a number of monochrome (absolutely black or absolutely white) blocks. Consequently, the earlier mentioned format for data storing becomes surplus and can be optimized. Evidently, it suffices to store (memorize) parameter values, specifying types of successive range blocks, and 3-tuples  $\langle e, f, t \rangle$  for non-monochrome (mixed) range blocks only. For instance, let us take a silhouette image  $256 \times 256$  (Fig. 3, b). Image content analysis shows that it contains 899 monochrome blocks and 125 non-monochrome blocks. Application of the fractal coding procedure (with the range blocks of size  $8 \times 8$  and the domain blocks of size  $16 \times 16$ ) leads to the following result – 4423 bits for stored data, and the image compression ratio reaches 14.82, i.e., turns out to be nearly five times better than in the previous case.

In general, amounts of stored data (compressed images) highly depend on both the internal structure of the image under processing and the size of the range (domain) blocks explored. In a particular case (for the bi-level image  $[X(m)]$  of size  $N \times N$ ), the number of bits for individual parameters, entering the earlier discussed format, can be defined this way:  $2 \cdot \log_2 N$  (bits), for the parameters  $e$  and  $f$ ; 2 (bits), for the parameter  $o$ ; 3 or 4 (bits), for the parameter  $t$ .

Finally, not going into details, we note that the fractal image decoding process starts with an initial image, comprising absolutely white and absolutely black blocks (depending on the stored values of the parameter  $o$  (Section 3.2)). Then the set of PIFS transforms (stored 3-tuples of data; Section 3.2) are repeatedly applied until the attractor is closely reached. The collage theorem tells us that this set of transforms, one for each range block, will have an attractor that is similar to the original image (Barnsley, 1993).

#### 4. Experimental Results

To corroborate theoretical analysis results obtained, a few many-sided experiments were carried out. The main goal in those experiments was to find out which one of the proposed two invariant image parameters (image smoothness index  $\alpha$ , or image coloration ratio  $\nu$ ; Section 2), being implemented into the fractal image encoding scheme (Fig. 1), gave the best results in the sense of image compression time savings and the quality of restored images. In connection with this, we here observe that no trials were made to compare efficiency of the proposed approach in the light of varying sizes of range (domain) blocks. Firstly, it is not a matter of great importance and, secondly (it goes without saying), the larger sizes of range (domain) blocks, the faster image encoding process and the lower quality of restored images (image estimates).

So, to carry out an experiment, quite a number of bi-level images, falling into different smoothness classes, were processed (Computer simulation was performed on a PC with CPU AMD1800+(@2800+), RAM 512MB, OS Windows XP), namely: *butterfly.bmp*, *girl.bmp*, *car.bmp*, *debate.bmp*, *face.bmp* and *dragon.bmp* (Fig. 2, 3).

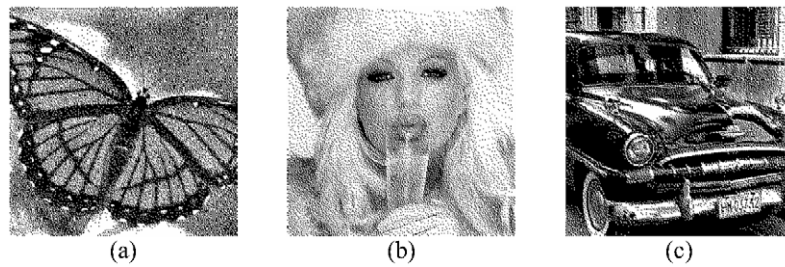


Fig. 2. Black and white images  $256 \times 256$ : (a) *butterfly.bmp*,  $\alpha = 0.536$ ; (b) *girl.bmp*,  $\alpha = 0.577$ ; (c) *car.bmp*,  $\alpha = 0.685$ .



Fig. 3. Black and white silhouette images  $256 \times 256$ : (a) *debate.bmp*,  $\alpha = 0.980$ ; (b) *face.bmp*,  $\alpha = 0.990$ ; (c) *dragon.bmp*,  $\alpha = 0.991$ .

All the selected images (Fig. 2, 3) were processed using two approaches: *Method 1* – detection of best pairings (range block – domain block) in the image is based on the use of image smoothness index  $\alpha$  (expression (6); Section 2); *Method 2* – detection of best pairings is based on the use of image coloration ratio  $\nu$  (expression (8); Section 2). Experimental analysis results (with range blocks  $U$  and domain blocks  $V$  of size  $8 \times 8$  and  $16 \times 16$ , respectively) are presented in Table 2.

As it can be seen (Table 2), better compression outcomes (for all images) were produced by *Method 2*. An essential detail – for non-silhouette images (*butterfly.bmp*, *girl.bmp*, *car.bmp*) the results, obtained using *Method 1* and *Method 2*, turned out to be terribly bad (the number of wrong pixels is very high), whereas, for silhouette images, both methods lead to quite reasonable results (the quality of restored images is good enough, time expenditures are tolerable). For instance, application of the totally exhaustive search (Table 3) and application of *Method 2* gave (for silhouette images) approximately the same result – the difference in the percentage of wrong pixels doesn't exceed 0.09% (the human eye is not sensitive to changes at such scale!). But, image compression time savings (in the sense of  $k = \tau_{total}^{\circ} / \tau_{total}$ ; Section 3.2) are more than obvious – vary from 165 to 475.

The serious conclusion can be made – the fractal image coding technique (strategy), in general, is inapplicable to bi-level images, but it is efficient enough in the case of silhouette images.

Table 2  
Preliminary experimental results (with the threshold value  $\varepsilon_0$  fixed)

Images	Method 1, $ \alpha_U - \alpha_V  \leq 0.013$		Method 2, $ \nu_U - \nu_V  \leq 1$		Compression ratio ( <i>bpp</i> )
	Wrong pixels (%)	Time (s)	Wrong pixels (%)	Time (s)	
<i>butterfly.bmp</i>	47.096	12.813	50.767	7.687	0.086
<i>girl.bmp</i>	27.136	13.703	27.136	2.797	0.089
<i>car.bmp</i>	56.109	13.734	30.017	9.422	0.064
<i>debate.bmp</i>	2.709	9.250	2.248	1.720	0.077
<i>face.bmp</i>	0.861	3.172	0.395	0.593	0.046
<i>dragon.bmp</i>	0.853	3.547	0.464	0.610	0.046

Table 3  
Totally exhaustive search (Jacquin's approach)

Experimental results	Silhouette images		
	<i>debate.bmp</i>	<i>face.bmp</i>	<i>dragon.bmp</i>
Wrong pixels (%)	2.167	0.378	0.374
Time (s)	285.189	282.164	287.523
Compression ratio ( <i>bpp</i> )	0.077	0.046	0.046

Also, efforts were made to find out what is the influence of the threshold value  $\varepsilon_0$  on the overall performance (percentage of wrong pixels and compression time expenditures) of the proposed fractal image coding strategy (Fig. 1). Two bi-level images – *butterfly.bmp* (Fig. 2, a) and *dragon.bmp* (Fig. 3, c) – were analysed. The results obtained confirmed once again that application of the fractal approach to encoding of non-silhouette images is deplorable (Table 4; range blocks and domain blocks were chosen to be of size  $4 \times 4$  and  $8 \times 8$ , respectively). As it can be seen (Table 4), for bi-level silhouette images, the best outcomes were fixed in the case of *Method 2*.

Referring to experimental analysis results, we observe that much more impressive fractal image compression speed gains can be obtained for bi-level images of size  $512 \times 512$ . For instance, for the silhouette image *madam.bmp*  $512 \times 512$  (Fig. 4; *Method 2*; range blocks and domain blocks were chosen to be of size  $4 \times 4$  and  $8 \times 8$ , respectively), the threshold value  $\varepsilon_0 = 0$  ensured very high image encoding time savings ( $k > 10^3$ ), moderate image compression ratio (0.049 *bpp*) and, above all, extremely high quality of the restored image – 0.808% of wrong pixels. For larger sizes of range blocks and domain blocks ( $8 \times 8$  and  $16 \times 16$ , respectively), application of *Method 2* to the same image gave the following results: compression time savings –  $k > 1.3 \cdot 10^3$ , image compression ratio – 0.017 *bpp*, quality of the restored image – 3.033% of wrong pixels.

Comprehensive analysis of these and many other bi-level images showed that the appropriately chosen threshold values  $\varepsilon_0$ , in many cases, paved the way for the detection of (nearly) optimal pairings in images under processing, as well as comparatively high overall performance of the proposed fractal image encoding strategy.

Table 4  
The influence of the threshold size on the bi-level image coding results

Images	Method 1			Method 2		
	Wrong pixels (%)	Time (s)	$\varepsilon_0$	Wrong pixels (%)	Time (s)	$\varepsilon_0$
<i>butterfly.bmp</i>	50.577	10.326	0.001	51.597	2.969	0
	47.849	12.625	0.01	50.767	7.687	1
	44.879	57.203	0.05	50.573	12.141	2
	44.206	91.641	0.1	50.233	20.797	4
	42.418	307.235	0.4	50.031	78.156	16
<i>dragon.bmp</i>	0.940	1.265	0.001	0.540	0.344	0
	0.864	2.250	0.01	0.464	0.610	1
	0.764	4.172	0.05	0.406	0.906	2
	0.661	7.766	0.1	0.368	1.422	4
	0.389	25.265	0.4	0.374	3.844	16



Fig. 4. Fractal bi-level image compression speed gains (silhouette image *madam.bmp*  $512 \times 512$ ; range blocks  $4 \times 4$ , domain blocks  $8 \times 8$ ).

## 5. Conclusion

In the paper, a new version of the fractal image coding strategy, adapted to process bi-level images, is presented. At the image encoding stage, the domain pools (limited search regions) for particular range blocks of the image under processing are formed using some newly introduced invariant image parameters (criteria), namely: (1) the total number of changes (in pixel values) along the image spatial axes – image smoothness index (expression (6); Section 2); (2) the relative number of “ones” and “zeros” in the image – image coloration ratio (expression (8); Section 2). Both criteria have been successfully employed to formulate the necessary image (block) similarity condition, which plays a key role in the developed strategy, i.e., in narrowing domain pools for the image range blocks. Experimental analysis results showed that both criteria did their work sufficiently well when applied to silhouette images. Exceptionally good performance was fixed in the case of the second criterion (image coloration ratio) – best pairings (range block – domain block) are searched among the image blocks containing comparatively the same number of “ones” (“zeros”). In parallels, a new format (structure) for storing of compressed bi-level images has been employed. An essential moment has been disclosed – application of the fractal approach to encoding of non-silhouette bi-level images is deplorable.

Finally, one more interesting detail has been found out – higher quality of restored bi-level images is obtained when (at the iterative image decoding stage) intermediate computational results are presented as real numbers (not integers), with the round up procedure at the very end.

In the future, some additional analysis, concerning the choice and substantiation of partitioning schemes in fractal bi-level image coding procedures, as well as justified applicability of the earlier developed image synthesis approach to fractal image decoding stage (Valantinas and Žumbakis, 2004), is supposed.

## References

- Bani-Eqbal, B. (1995). Enhancing the speed of fractal image compression. *Optical Engineering*, **34**(6), 1705–1710.
- Barnsley, M.F. (1993). *Fractals Everywhere*. Academic Press Professional, Cambridge.
- Blackledge, J.M., A.K. Evans and M.J. Turner (2004). *Fractal Geometry: Mathematical Methods, Algorithms, Application*. Albion/Horwood.
- Dimri, V.P. (2000). *Applications of Fractals in Earth Sciences*. Taylor&Francis, London.
- Evans, A.K. (1997). Fourier dimension, fractal dimension and the fractional derivative. *Technical Report*, **23**, SERCentre, De Montfort University, Leicester LE 1 9BH.
- Fisher, Y. (1994). *Fractal Image Compression – Theory and Application*. Springer-Verlag, New York.
- Furao, S., and O. Hasegawa (2004). A fast no search fractal image coding method. *Signal Processing: Image Communication*, **19**(5), 393–404.
- Hamzaoui, R. (1997). Codebook clustering by self-organizing maps for fractal image compression. In *NATO ASI Conf. Fractal Image Encoding and Analysis*, vol. 5. Trondheim (supplementary issue).
- Hamzaoui, R., and D. Saupe (2000). Combining fractal image compression and vector quantization. *IEEE Transactions on Image Processing*, **9**(2), 197–208.
- Hassaballah, M., H.M. Makky and Y.B. Mahdy (2005). A fast fractal image compression method. *Electronic Letteres on Computer Vision and Image Analysis*, **5**(1), 30–40.

- Jacquin, A. (1992). Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transactions on Image Processing*, **1**(1), 18–30.
- Kaandorp, J.A., and J.E. Kuebler (2001). *The Algorithmic Beauty of Seaweeds, Sponges and Corals*. Springer-Verlag, Heidelberg, New York.
- Peitgen, H.-O., H. Jurgens and D. Saupe (1992). *Chaos and Fractals*. Springer-Verlag.
- Polvere, M., and M. Nappi (2000). Speed-up in fractal image coding: comparison in methods. *IEEE Transactions on Image Processing*, **9**(6), 1002–1009.
- Rao, K.R. (1985). *Discrete Transforms and Their Applications*. John Wiley&Sons, Incorporated, New York, NY, U.S.A.
- Samko, S.G., A.A. Kilbas and O.I. Marichev (1993). *Fractional Integrals and Derivatives: Theory and Applications*. Gordon and Breach Scientific Publishers, Switzerland Philadelphia, Pa., USA.
- Saupe, D., and R. Hamzaoui (1994). Complexity reduction methods for fractal image compression. In J.M. Blackledge (Ed.), *Proceedings of the IMA Conference on Image Processing: Mathematical Methods and Applications*. Oxford, England. pp. 211–229.
- Saupe, D. (1995). Accelerating fractal image compression by multidimensional nearest neighbour search. In J.A. Storer and M. Cohn (Eds.), *Proceedings DCC'95 (IEEE Data Compression Conference)*. Snowbird, UT, USA. pp. 222–231.
- Saupe, D. (2002). Fractal image compression via nearest neighbour search. *Fractals in Multimedia, The IMA Volumes in Mathematics and its Applications*, vol. 132. Springer Verlag.
- Tong, C.S., and M. Pi (2001). Fast fractal encoding based on adaptive search. *IEEE Transactions on Image Processing*, **10**(9), 1269–1277.
- Tong, C.S., and M. Wong (2002). Adaptive approximate nearest neighbor search for fractal image compression. *IEEE Transactions on Image Processing*, **11**(6), 605–615.
- Truong, T.-K., J.-H. Jeng, I.S. Reed, P.C. Lee and A.Q. Li (2000). Fast encoding algorithm for fractal image compression using the DCT inner product. *IEEE Transactions on Image Processing*, **9**(4), 529–535.
- Turner, M.J., J.M. Blackledge and P.R. Andrews (1998). *Fractal Geometry in Digital Imaging*. Academic Press, Cambridge.
- Valantinas, J., N. Morkevičius and T. Žumbakis (2002). Accelerating compression times in block-based fractal image coding procedures. In *Proceedings of the 20th Eurographics UK Conference*. Leicester (De Montfort university, UK), IEEE Computer Society Press (Los Alamitos, California). pp. 83–88.
- Valantinas, J., and T. Zumbakis (2004). On the use of shift dynamics in synthesizing fractal images. *Informatica*, **15**(3), 411–424.
- Valantinas, J., and T. Žumbakis (2004). Definition, evaluation and task-oriented application of image smoothness estimates. *Information Technology and Control*, **1**(14), 15–24.
- Valantinas, J., and T. Žumbakis (2005). The Use of Image Smoothness Estimates in Speeding Up Fractal Image Compression. In *Proceedings of the 14th Scandinavian Conference SCIA 2005*, Joensuu, Finland. *Lecture Notes in Computer Science*, vol. 3540. Springer-Verlag, Berlin Heidelberg. pp. 1167–1176.
- Wohlberg, B., and G. de Jager (1999). A review of the fractal image coding literature. *IEEE Transactions on Image Processing*, **8**(12), 1716–1729.
- Yokoya, N., F. Yamamoto and N. Funakubo (1989). Fractal based analysis and interpolation of 3d natural surface shapes. *Computer Vision Graphics and Image Processing*, **46**, 284–302.

**J. Valantinas** received the PhD in 1980. He is a professor at the Department of Applied Mathematics in the Kaunas University of Technology (Lithuania). His major research interests include discrete transforms, digital image processing and fractal technologies.

**T. Žumbakis** is currently a PhD student in the Department of Applied Mathematics of Kaunas University of Technology (Lithuania). He received the MS degree at the Computer Engineering Department from the Kaunas University of Technology in 2002. His major research interests include data compression and fractal image encoding technologies.

## **Invariantinių vaizdo parametrų taikymas dvispalvių vaizdų fraktalinio kodavimo procedūrose**

Jonas VALANTINAS, Tomas ŽUMBAKIS

Fraktalinis vaizdų suglaudėjimas yra viliojanti ir perspektyvi technologija, kuri gali būti sėkmingai panaudota statiniams vaizdams koduoti, ypač esant aukštam informacijos suspaudimo laipsniui. Pagrindinė kliūtis, kurią reikia įveikti – didelės vaizdo kodavimo etapui tenkančios laikinės sąnaudos. Nepaisant skaitlingų ir įvairiapusių pastangų, siekiant pagreitinti fraktalinį vaizdo kodavimo etapą, „greičio problema“ iki šiol nėra pilnai išspręsta.

Straipsnyje pateikiama nauja modifikuota fraktalinio vaizdų kodavimo strategijos versija, orientuota apdoroti dvispalviams (juodai-baltiems) vaizdams. Siūloma strategija remiasi būtina vaizdų panašumo sąlyga, kuri, savo ruožtu, grindžiama invariantinių vaizdo parametrų (vaizdo glodumo indeksai, vaizdo spalviniai santykiai) taikymu. Parodoma, jog vaizdai negali būti panašūs (vidutinės kvadratinės paklaidos prasme), jeigu jų atitinkamų parametrų reikšmės kiek žymiau skiriasi. Siūlomoje strategijoje būtina vaizdų panašumo sąlyga vaidina svarbų vaidmenį – ji panaudojama optimalių porų (reikšmių srities blokelis–apibrėžimo srities blokelis) paieškos vaizde procesui greičinti, t.y., leidžia sumažinti apibrėžimo srities baseiną (paieškos sritį) kiekvienam reikšmių srities blokeliui. Eksperimentinės analizės rezultatai rodo, jog naujos fraktalinio vaizdų kodavimo strategijos panaudojimas žymiai sumažina dvispalvių vaizdų fraktalinio kodavimo laikines sąnaudas. Išskirtinai geri kodavimo rezultatai (kodavimo laikai, atkurtų vaizdų kokybė) siejami su siluetiniais vaizdais.