Feature generation from digital images using pseudo-fractal algorithm and its four modifications

Marcin Janaszewski, Edward Kacki
Department of Expert Systems and Artificial Intelligence,
The College of Computer Science, Rzgowska 17a, 93-008 Lodz, Poland

January 5, 2007

Abstract. The main plot of the paper is to present the authors' original method of feature generation from digital images and to report on comparison of five various algorithms, which realise the method. The algorithms are based on the same authors' idea, consists in quantitative description of similarity intensity between various parts of an image in various scales. To do so the algorithms take advantage of fractal coding based on an Iterated Function System. Therefore generated features can rightly be called similarity features. In this paper we show that similarity features, when combined with other well known ones, can improve recognition results in some image classification tasks. After the presentation of the algorithm way of working, we compare their properties and report classification results obtained in two different pattern recognition experiments. Moreover, the paper contains a discussion of obtained results, and possible future applications of similarity features.

Key words: pattern recognition, features generation, texture analysis, self-similarity

1. Introduction

One of the most intensively investigated issue of computer science is currently automatic recognition of patterns. The problem is of great importance due to wide application of recognition systems in many fields of human activity. The systems could assist the human in most situations, in which a decision is made based on image analysis. For example, such a system could be applied to images characterized by strong self-similarity, like some medical textures, e.g. bone tissue images. The system would make it possible to discriminate between images representing pathological and normal objects basing on weakening of self-similarity properties in pathological objects. Another application could be an image retrieval system, which performs feature generation from a large set of images and creates feature indices. In retrieval, the system extracts features of a query image and compares them to features of classes of objects rather than features of individual objects. Such a system retrieves images containing instances of a class of objects rather than images, which contain a specific object [11].

The fundamental problem in pattern recognition is generation of a small set of discriminatory features representing an object or an image. The set is called feature set. In general we do not build classification systems, which processes whole images. Such sys-

tems would work very slow, and would fail in classification due to a lot of unimportant, distracting information which occurs in images. Instead, we try do describe images with a small set of important features, which makes up input data. The input features should strong discriminate classes, be invariant with respect to geometric and photometric image deformations. Moreover they should be easy to interpret and robust, which means that small changes of an image result in small changes of its features. Finally generation procedure should exploit possibly little computational resources.

The image analysis theory does not provide any significance evaluation methods capable of operating on a set of features that have not been extracted before. Therefore, a system constructor is frequently forced to extract a large set of various features, from which significant ones are chosen with the help of known methods[20]. It should be emphasized that the process of extraction of significant features is largely dependent on individual properties of objects being examined and on the character of the recognition task. There is, therefore, an impressive number of publications, e.g. [21][17][28][15][18][19], presenting search for sets of significant features adequate for tasks to be solved and enabling effective pattern recognition. Thus system designers can either make use of the entire arsenal of features, cf. [15][20], whose usefulness has successfully been proved in the tasks already solved, or to venture to create new feature generation algorithms with the aim of constructing systems that would be still more effective or helping to deal with recognition tasks that are still waiting for the computer-assistance to be solved.

Despite many years of research, there are still some domains where no success in creating systems capable of analyzing all classes of images of a given type has been won. Such is the situation, for instance in texture analysis where, in spite of intensive investigations targeted at assessment of discrimination parameters of textures, e.g. [23][17][28][22][32], no complete quantitative description that would enable texture discrimination of all classes has been built.

Image encoding based on an Iterated Function System (IFS) is well known in literature and since proposed for the first time by Bansley [3], has been widely used for image compression. The most important improvement of Bansley algorithm was proposed by Jaquin [4] who introduced additional division of an image in order to simplify automatic search for self-similar region. Since than we can observe that most effort has been done to improve IFS based compression algorithms e.g. [7][30][13][16]. There are also many papers in which authors attempt to apply IFS coding to image classification, segmentation or retrieval e.g. [21][10][29][25][27]. Nevertheless, we have found only several works, which are relatively close to our approach [21][25][27]. We present the general description of them in the next Section.

In the view of the above observations, it seems desirable to conduct research on construction and application of new feature generation methods which involve IFS. The authors attempt to examine discriminative capabilities of the features describing self-affinity intensity in an image. The algorithms presented in the paper quantitative de-

scribe intensity of self-affinity in an image being tested. The self-affinity intensity can be roughly defined as an average similarity of affine transformed fragments of an image to different fragments of the image. It seems that actually an image can be investigated with the aim of determining the degree with which a visualized object possesses a given feature. If among objects of the same type there occurs a feature definable by strong self-affinity, then pathological objects of the same type might be identified by apparent attenuation of this feature.

2. Fractal coding

Our research is inspired not only by successful fractal compression of images but also by application of fractal theory to quantitative description of objects' shapes. Generally, it is hardly possible to find an optimal fractal code for an image except for a subset of images that are proper fractals in the mathematical sense [1]. Indeed, this problem is NP-hard[13]. Although Barnsley, [2][3][8], suggested an efficient technique for fractal code approximation applicable to any image, this is not the approximation method that is capable of solving the optimization problem posed in [13]. The fractal image coding makes use of similarity of image fragments to determine a set of contractive transformations constituting a fractal transform of the image. Although theoretically any contractive transformations would make fractal code of an image, in practice affine transformations are used.

The determined transformations applied repeatedly to any initial image lead to the coded image reproduction burdened with a small error. According to Collage's theorem[8], for any subset E of space R^n and for any approximation precision, it is possible to find an IFS that can be used to approximate set E with a desired precision.

Detailed description of fractal image compression is presented elsewhere [7]. Therefore we only want to give general view of the method because it is strictly connected with our approach.

The fractal code of an image consists of a number of affine transformations w_i of domain regions D_i into range regions R_i:

$$R_i = w_i(D_i) \tag{1}$$

Both R_i and D_i represent fragments of an image being transformed (sets of pixels described by three values, two spatial coordinates and brightness). An affine transformation w_i may be defined in the following way:

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix} \tag{2}$$

where x, y represent co-ordinates of a pixel from D_i and z is a brightness level of the pixel. a_i, b_i, c_i, d_i, express a combination of rotations, scalings and shears while e_i, f_i represent translations. s_i is a contraction coefficient of grey levels and o_i represents translation of grey levels.

In classical fractal compression approach not all possible affine mappings are allowed. The set of possible affine mappings is determined by special procedure of domain and range selection from an image. In the simplest approach, ranges are chosen by quaternary division of an image, which results in a set of squared nonoverlaping fragments. Domains are selected by gradual squared frame shifting per one pixel. In addition domains are rotated by 90, 180, 270 and reflected about x, y axis and diagonals. All domains and their rotated and reflected versions make up the set of domains DS. The fractal compression procedure searches, for each range, a similar domain from DS. The procedure exploits last square fit to calculate, for each pair domain-range, the best grey levels contraction s_i and grey levels translation o_i.

Our algorithm of feature generation is very similar to fractal compression procedure. The main difference lies in the different goals of the algorithms. The fractal compression algorithm evaluates fractal code of an image while our procedure only quantitatively describes intensity of similarity between fragments of an image.

In spite of the fact that there are many papers that present applications of IFS in image analysis, we have found only several papers, which are close to our method. In [21] Baldoni et al. propose the method of fractal feature generation in the form of the most important affine transforms (2) of the whole image to its fragments. The method is designed to images which represent isolated objects and was tested in two different experiments: hand-written digits recognition and classification of artificially constructed images of tress. Taking into account the small number of features exploited in the experiments, the authors achieve quite good recognition results, in comparison with different approaches.

In [27] Yokoyama et al. propose image retrieval system using IFS codes. They treat translation vectors between similar regions of an image as features of the image. Than defining the distance between two sets of features, they are able to find, in the image base, several candidates which are very similar to a query image in the sense of the distance.

The fractal code decoding process is always convergent to the same image, called an attractor, independently of the starting image. For a given fractal code, an attractor is always a unique image. This fact was used for face identification by Tan and Yan [25] who applied fractal codes of images x_i, $i = 1, 2, \ldots, N$, from database and a decoding process of the fractal code to identify an unknown face image x. The identification process included creation of N new images y_i, i = 1, 2,. . . ,N, through the decoding of codes of images x_i from the database using x as a starting image, and calculation of N Euclidean distances $d(x_i, y_i)$. Finally, x was identified as xi if and only if $d(x_i, y_i)$ was

the value minimal among $d(x_k, y_k)$, $k = 1,2,\ldots,N$. It may be interesting that practically in the presence of geometric and photometric deformations of the image the results achieved by means of this method proved much better than those provided by other face identification procedures.

3. Pseudo-fractal feature generation algorithm

The pseudo-fractal feature generation algorithm (PFA), is our original idea of feature generation, based on compression method presented in [7], modified with respect to the operational speed and the aim of finding self-affinity. PFA can be applied to square images with sides the length of which, expressed in pixels, is a power of 2. From now till the end of the paper, an image is treated as a matrix whose element in the i–th row and j–th column includes the grey level, in the range ¡0; 255¿, of the pixel situated in the i–th row and j–th column of the image. An essential element of the algorithm discussed here, i.e. an operation for which the name "similarity testing" has been coined, can be presented in the following few stages:

1. Isolation of two subimages from the examined image, one of which is called range (R) and the other is referred to as domain (D). The domain is assumed not to be smaller that the range. In addition, R and D stand for matrices representing the map of the range grey color levels and that of the domain ones, respectively.
2. Diminishing of domain D to the range size (only if the domain is larger than the range), by averaging of the neighboring pixels. In this way, a new matrix D' is obtained.
3. Determination, by means of the least square method, of the best affine transform represented by two real values $(\bar{x}, \bar{y})$. To do so, the definition of a new function $g : \mathbb{R}^2 \to \mathbb{R}$ is introduced:

$$g(a, b) = \|aD' + b - R\| \tag{3}$$

where: $\| * \|$ – is a norm defined as:

$$\forall A = [a_{ij}]_{\substack{i=1,2,\ldots,n \\ j=1,2,\ldots,m}} : \quad \|A\| = \frac{1}{|A|} \sum_{i=1}^{n} \sum_{j=1}^{m} |a_{ij}| \tag{4}$$

$|A|$ – number of elements in matrix A
operation $aD' + b$ denotes multiplication of each matrix D' element by real value a and addition of real value b to each element of matrix aD'.

Than values $\bar{a}, \bar{b}$ are determined by minimisation of g function:

$$g(\bar{a}, \bar{b}) = \min_{(a,b)\in\mathbb{R}} g(a, b) \tag{5}$$

In order to minimise calculations, the least square algorithm take advantage of the range and the domain histogram only if the range size is larger than 256 pixels.

4. Calculation of similarity error $Err(R, D)$:

$$Err(R, D) = \frac{1}{|R|} \left\| R - \bar{a}D' + \bar{b} \right\| \tag{6}$$

Similarity error $Err(R, D)$, which is a similarity testing result, describes a degree of similarity, in the sense of affine transformation, between domain D and range R.

Stage 1, and in particular the method of the range and domain choice, may need some more detailed explanation. The range selection is carried out by means of a quaternary division of an image. The first two hollows of the division and the way of range numbering are shown in Fig 1. The symbols introduced here should be understood as follows:

R_{ik} - k-th range in the i-th hollow, $i = 1, 2, \ldots, \log_2(is) - 1$, $k = 1, 2, \ldots, 4^i$;

Each range in the i-th hollow has the side equal to $(is)/2^i$, where:

is – size of an image side, measured in pixels

$R_{ik} \in \boldsymbol{R_i}$; $\boldsymbol{R_i}$ – set of all ranges in hollow i.

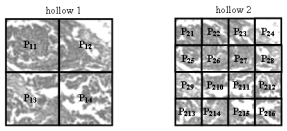

Fig. 1. The first two hollows of quaternary division of an image

The selection method for domains, presented for a square image of 8 pixel side, is illustrated in Fig 2. Each blank represents one pixel of the image under testing. Successive domains are chosen by gradual frame shifting per 1 pixel. The D_{jl} symbol stands for the l-th domain in the j-th hollow. $j = 1, 2, \ldots, \log_2(is) - 1$;
$l = 1, 2, \ldots, ((is - ds_j) + 1)^2$; where:

is - size of the image side in pixels,

ds_j - size of a domain side in the j-th hollow, measured in pixels. $ds_j = is/2^j$

$D_{jl} \in \boldsymbol{D_j}$; $\boldsymbol{D_j}$ – set of all domains in j hollow.

When hollow of the domain and the range are equal ($i = j$), then similarity testing is performed only for the domain-range pairs that have no common pixels, i.e. when the part of the image being the domain does not overlap the part being the range. Feature C_{ij} of the tested image is formed in two stages:

1. For each range $R_{ik} \in \boldsymbol{R_i}$ the algorithm calculates E_{ijk} basing on the following formula:

$$E_{ijk} = Err\left(R_{ik}, \bar{D}_j\right) \quad \bar{D}_j \in \boldsymbol{D_j} \tag{7}$$

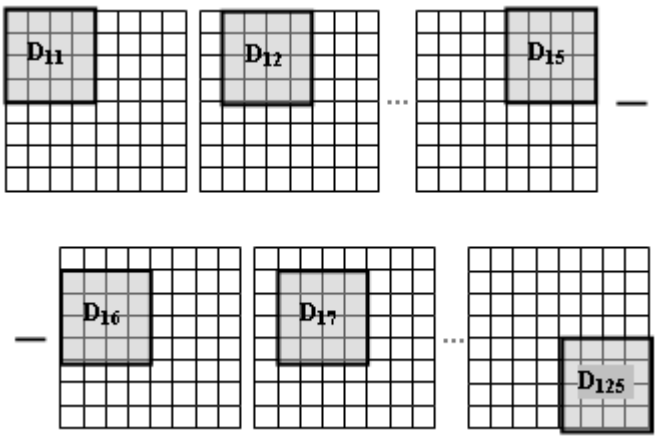

Fig. 2. Domain choice, shown for the first hollow of a square image of 8 pixel size.

where $\bar{D}_j$ is the first domain, from the $\boldsymbol{D_j}$ set, found to fulfil the condition:

$$E_{ijk} = Err\left(R_{ik}, \bar{D}_j\right) \leq AErr \tag{8}$$

in which $AErr$ is an algorithm parameter and denotes an acceptable error.
If, in set $\boldsymbol{D_j}$, there is no domain that fulfils condition (8), then:

$$E_{ijk} = \min_{D \in \boldsymbol{D_j}} Err\left\{R_{ik}, D\right\} \quad \text{for } k = 1, 2, \ldots, 4^i \tag{9}$$

2. Feature calculation

$$C_{ij} = \frac{1}{|\boldsymbol{R_i}|} \sum_{k=1}^{|\boldsymbol{R_i}|} E_{ijk} \tag{10}$$

for: $i = 1, 2, \ldots, \log_2(is) - 1$; $j = 0, 1, \ldots, i$;
where: $|\boldsymbol{R_i}|$ – the number of elements in $\boldsymbol{R_i}$ set.
is – size of an image side, measured in pixels

Domain D satisfying the condition: $Err(R, D) < AErr$ is said to be similar to range R.

In order to accelerate feature extraction, we invented and implemented an original module, called domain preparation (DP). The module is designed to prepare auxiliary images from which grey levels can be identified, which in turn enables avoidance of diminishing of each domain separately. To explain how the module operates let us

consider the square image of 8 pixel side and present the method of constructing auxiliary images applied to it. In Fig 3 that illustrates the discussed procedure, the little white squares represent pixels of an image under examination, while the black ones, formed by the union of four neighbouring white squares, stand for the pixels of the created auxiliary images: O_{100}, O_{101}, O_{110}, O_{111}. As the first digit in the subscript denotes the difference in the hollows of the ranges and the domains, therefore in the case under consideration, the difference equal to 1 implies that auxiliary images are prepared for the domains that are twice as big as the ranges. The remaining two digits in the subscript denote the address of the first pixel of the examined image taking part in the generation of the auxiliary image. Let us assume now that in an image with sides equal to 8, the range numbered 0 in hollow 2 and the domain numbered 7 in hollow 1 are to be compared. Due to calculations carried out in module DP the domain need not be diminished twice by averaging four neighbouring pixels of the examined image. It suffices to read appropriate pixels in image O_{111}, more precisely, the first pixel of this image and the analogous pixels of its neighbours. It is worth pointing out that the application of module DP shortens the operation time of the algorithm by about 20 per cent.

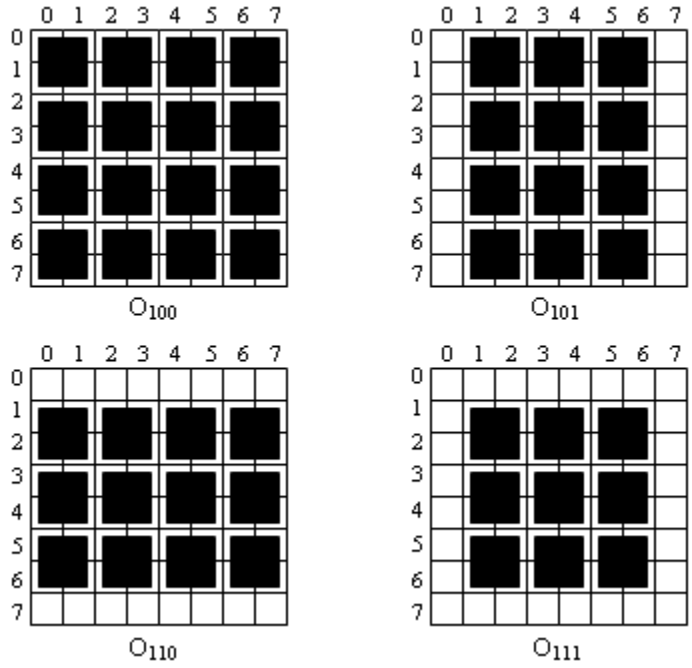

Fig. 3. Auxiliary image construction, for square image of 8 pixel side, in Domain Preparation module

Finally we wrote an algorithm, which includes DP module and extracts a set of similarity features (SFs). The algorithm input parameters are: an image, $Nmax$ – maximal hollow in image division, $Nmin$ – minimal hollow in image division, $AErr$ – acceptable error. The algorithm extracts features C_{ij} for range hollows $i = Nmin, Nmin + 1, \ldots, Nmax$ and domain hollow $j = i, i-1, \ldots, Nmin - 1$. For example applying $Nmin = 3$ and $Nmax = 5$ we obtain the following set of features: C_{33}, C_{32}, C_{44}, C_{43}, C_{42}, C_{55}, C_{54}, C_{53}, C_{52}.

When testing PFA in several pattern recognition experiments, we were able to find a few interesting properties of both the algorithm and the similarity features. It was also shown, however, that because of an excessively long time of features extraction, the applicability of the algorithm became limited to the square images of side smaller than 256 pixels. This obvious drawback strengthened the need for modification of PFA so that the similarity features extraction process could be accelerated. The properties of PFA and similarity features are presented in the next Section whereas the modifications of PFA can be found in Section 5.

4. Properties of similarity features

One of the characteristics of the PFA is that although it is designed for examination of self-affinity, no image model whatsoever needs to be assumed. The only assumption that is made is that the degree of the similarity intensity characteristic of certain image classes is a discriminating property for these classes. The lack of the fractal model assumption is of special significance for many types of images, which in most cases exhibit neither strict nor statistic self-affinity.

Another important characteristic of PFA is that it investigates self-affinity separately in various scales, i.e. self-affinity of parts of different sizes can be examined. For example, for a square image of 64 pixel side, the following features may be extracted:

$C_{64,32}$ – defining self-affinity degree between the whole image and its square fragments of size 32×32 pixels.

$C_{64,16}$ – defining self-affinity degree between the whole image and its square fragments of size 16×16 pixels.

...

$C_{64,2}$ – defining self-affinity degree between the whole image and its square fragments of size 2×2 pixels.

$C_{32,32}$ – defining self-affinity degree between image fragments of size 32×32 pixels and other fragments of the same size such that the common part of the fragments is empty.

$C_{32,16}$ – defining self-affinity degree between image fragments of size 32×32 pixels and those of size 16×16 pixels.

...

$C_{32,2}$ – defining self-affinity degree between image fragments of size 32×32 pixels and those of size 2×2 pixels.

...

$C_{2,2}$ – defining self-affinity degree between image fragments of size 2×2 pixels and other fragments also of size 2×2 pixels and such that the common part of the fragments is empty.

Consequently, for accurate recognition it suffices for the examined objects of various classes to have significantly different feature values for one of the divisions (one feature within C_{ij}); it is worth pointing out that no strict or statistic self-affinity of the objects is necessary.

An advantage of similarity features (SFs) following directly from the nature of the affine transformation is their invariability with respect to affine transformations of grey levels of image pixels. Hence, the features are resistant to changes of the lighting conditions to which created images may be subjected (linear changes of contrast or lighting intensity). Unfortunately, because of the quaternary image division, not in every situation are SFs invariant with respect to geometric image deformations. Nevertheless, in certain situations they exhibit resistance to shifting, scaling and rotation of a whole image or part of it. The situations under consideration may be classified in the following way:

1. An image is deformed so that the parts of the original image comprised in similar ranges and domains are not transferred outside these areas. This sort of the deformation resistance is dependent on the range size and the type of the image, especially on the local properties of the image in the neighbourhood of two similar subimages, i.e. domain and range. The SFs independence of this sort of deformation is illustrated in Fig 4. Fig 4 a) shows an original image with a grid of ranges marked on it and two pairs of similar subimages, R_1, D_1, and R_2, D_2. Fig 4 b) displays the same image on which slight shifting, rotation and scaling have been performed. It turns out that despite the deformation, the marked pairs of subimages still remain similar.

2. An image is deformed so that at least one part of the original image situated in range R or domain D, where R and D are assumed to be similar, is transferred outside the area. Depending on the image type and the transformation applied, the similar elements can be transferred into an area of another domain D' and range R' substituting the similar pair R, D with pair R', D'. Such is the situation presented in Fig 5.

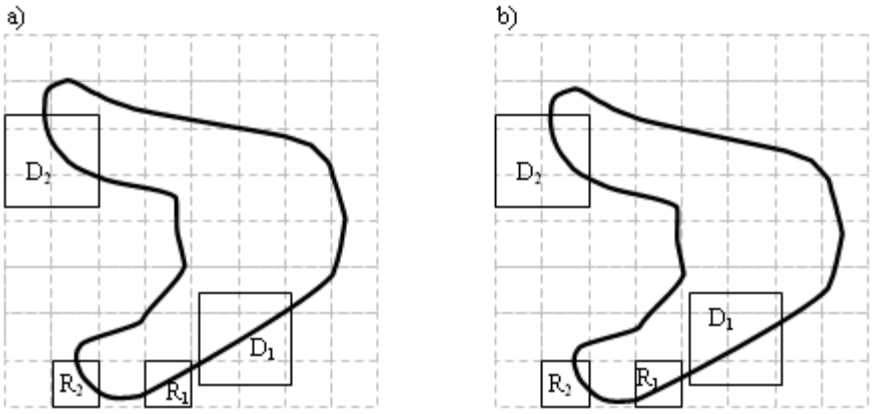

Fig. 4. Similarity independence of pairs R_1, D_1 and R_2, D_2 of the image curve deformation through scaling, rotation, and shifting. a) Original image with indicated areas of similar domains and ranges. b) Image deformed with respect to the original through rotation, scaling, and shifting in which indicated domain and range areas are the same as those on the left.

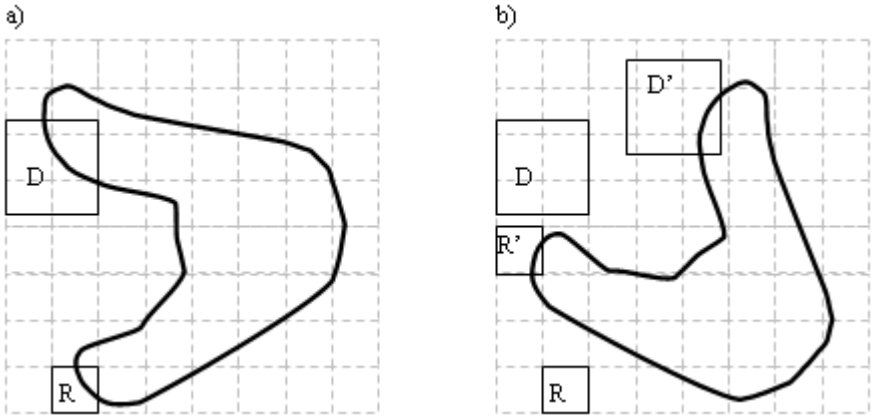

Fig. 5. SF independence of essential geometric image deformations. a) Original image with indicated areas of similar domain D and range R. b) Image considerably deformed with respect to the original through rotation, scaling, and shifting in which indicated domain and range areas are the same as those on the left.

5. Modifications of pseudo-fractal feature generation algorithm

5.1. Pseudo-fractal algorithm with multistage shifts

The modification of PFA leading to pseudo-fractal algorithm with multistage shifts (PFAMS) lies in a change in the domain selection for the similarity test by application of the multistage shift mechanism.

PFAMS construction was inspired by the observation that in many images the prevailing number of changes are situated in the low or middle-frequency bands, which means that generally grey levels of neighbouring pixels do not differ much. This allows the conclusion that in such images neighbouring domains do not differ much, either.

Hence it is possible to extract SFs by testing every second or every fourth domain. Such an approach results in the acceleration of the features extraction process, and at the same time the extracted features still represent the degree of similarity between different parts of the image. At the first stage of PFAMS, every sixteenth domain is tested, which means that the frame is shifted per 16 pixels. The features extracted at the first stage are very likely to be recognisable. If, however, the features turn out to have low class discrimination power, one can take advantage of the second stage at which every eighth domain is chosen, i.e. the frame is shifted per 8 pixels. If, again, the features are of low class discrimination power, the third stage, where every fourth domain is tested, can be applied. Continuing, at the next stage every second domain is tested, and at the last (fifth) one the frame is shifted per one pixel, exactly like in PFA. Moreover, the features extraction process may be additionally accelerated thanks to the maximal use of the data accumulated at the preceding stages [26].

5.2. Pseudo-fractal algorithm with initial classification

When testing PFA in several pattern recognition experiments, we noticed that in order to find a similar domain, for a range, the algorithm has to examine a lot of domains, which are not similar to the range. The domains while testing give high values of similarity error. Therefore it would be good idea not to examine domains which have no chance to be similar to particular range. To do so we extended the PFA by adding the procedure called initial classification (IC). In this way we constructed pseudo-fractal algorithm with initial classification (PFAIC). The IC procedure aim of working is to establish pairs of image fragments which are suspended to be similar. To do so the procedure calculates for each domain D three parameters of the domain grey levels: mean value mD, standard deviation dD and skewness bD. Then for each range R corresponding parameters are calculated: mR – mean value, dR – standard deviation and bR – skewness. Finally the procedure constructs for each range R, a set D_{Rj} of domains from j depth, defined in

the following way:

$$D_{Rj} = \{D \in \boldsymbol{D_j} : \quad |mR - mD| < mMax \wedge |dR - dD| < dMax \wedge \\ \wedge |bR - bD| < bMax\} \tag{11}$$

where: $mMax$, $dMax$, $bMax$ - are parameters of the IC procedure given by the user.

If domain D belongs to D_{Rj} than D is called accepted domain. If IC procedure sets a domain to be an element of at least one D_{Rj} than it is called that the procedure accepts the domain. Once all D_{Rj} sets are constructed, the algorithm, for each range R, searches through D_{Rj} for the similar domain.

The classes discriminatory power of features extracted by PFAIC depends, to a large extent, on $mMax$, $dMax$, $bMax$ parameters. The perfect situation is when the parameters are set in such a way that, for each range R and domain depth j, D_{Rj} sets include only domains, which are really similar to the range R.

Unfortunately, because of the limited accuracy of image description based on the three parameters described above, there is little probability to obtain such a perfect situation in practical applications. Nevertheless the parameters should satisfy the following postulate:

The IC procedure, for the sake of execution acceleration, should accept as few domains as possible. On the other hand, for the sake of the best possible description of an image similarity, it should accept as many similar domains as possible.

Intuitive selection of $mMax$, $dMax$ and $bMax$ parameters, according to the postulate, is for sure very difficult, even for a user who is experienced in application of PFA and PFAIC. Therefore we constructed an optimization algorithm, which automatically sets the parameters[26].

5.3. hybrids of pseudo-fractal algorithm with multistage shifts and pseudo-fractal algorithm with initial classification

It seems that interesting results can be obtained for an algorithm, which is a hybrid of PFAMS and PFAIC. Such an algorithm can be the fastest of all presented pseudo-fractal algorithms, but it's speed, because of initial classification application, is strongly dependent on the image contents.

The idea presented above was the basis for construction of two multistage algorithms, which use two procedures to restrict the set of domains designed for similarity testing. The first of the algorithms – pseudo-fractal algorithm with multistage shifts and initial classification (PFAMSC) can be run at one of five stages, like PFAMS. At the first stage of PFAMSC, for each domain depth j, a new set $\boldsymbol{D_{j1}}$ is constructed. $\boldsymbol{D_{j1}}$ consists of every sixteenth domain form a set of all j depth domains $\boldsymbol{D_j}$. Than $\boldsymbol{D_{j1}}$ is restricted by initial classification procedure, resulting in $\boldsymbol{D_{j1C}} \subset \boldsymbol{D_{j1}}$ set. Finally, only domains

from $\boldsymbol{D_{j1C}}$ take part in similarity testing procedure. Similarly, at the second stage each $\boldsymbol{D_{j2}}$ set consists of every eight domain form j depth and each $\boldsymbol{D_{j2C}}$ is the result of initial classification procedure applied for $\boldsymbol{D_{j2}}$. Finally, at the fifth stage $\boldsymbol{D_{j5}} = \boldsymbol{D_j}$ and PFAMSC generates the same set of features as PFAC.

The second idea - pseudo-fractal algorithm with initial classification and multistage shifts (PFACMS) is very similar to PFAMSC. The only difference lies in the fact that PFMCMS on each of five stages first use initial classification algorithm and than multistage shift procedure.

6. 6. Experiments in self-affinity feature recognition

6.1. Textures

To compare the recognition accuracy of SFs with that of other known features extracted from images representing textures, we applied visualization of artificial objects (physical models – phantoms). The phantoms were visualized by means of a magnetic resonance tomograph at the Medical Physics Department, University of Dundee, Scotland, to be then used as research material for COST B11 project[15]. In this way two 8-bit images represented scans of cross-section of two various-porosity foams were obtained. From each optical image, 16 non-overlapping samples of size 32×32 pixels were taken, resulting in 2 texture classes, each of 16 samples.

From the images, a total of about 350 SFs were extracted – 70 features with each pseudo-fractal algorithm. Moreover, in order to compare class discrimination power of SFs with class discrimination power of typical features commonly used in texture recognition tasks, more than 280 features were extracted by means of MaZda Program [31][20] (MaZda - the abbreviation for "Acierz ZDArzen"- in Polish). MaZda, among other things, enables feature generation [31][20] basing on: image histogram, image gradient, run-length matrix, co-occurrence matrix for four directions and five distances between image pixels, autoregressive model.

In what follows, features extracted using MaZda program will be in short referred to as MaZda features. MaZda features makes up standard quantitative description of textures extensively used in many texture recognition tasks. Subjecting the features to transformation by means, for instance, of PCA[20] or Fisher discriminative analysis [20], it is possible to obtain a small set of weakly correlated features that strongly enough discriminate the object classes. We took advantage of floating forward feature selection [20] and multidimensional Fisher coefficient as a class separation criterion. The Fisher coefficient is defined as the ratio of mean-squared between-class distance (computed between the class means m_k, $k = 1, 2, \ldots, K$; K – number of classes) to the mean of mean-squared within-class distances (computed between the samples of class k and the corresponding class mean m_k for $k = 1, 2, \ldots, K$). Therefore the greater value of F the

stronger classes discriminatory power of the feature set to be examined.

In order to carry out the recognition task, eleven testing sets, described in Table 1, were constructed. Once we had performed features selection procedure it turned out that each of the following sets: M&PFT, M&MST, M&ICT, M&MSCT, M&CMST consisted of one MaZda and one SF.

To recognize the extracted features the following methods have been exploited: k–nearest neighbour (k–NN)[5], nearest mode (NM)[5], multilayer perceptron MLP[9], neuro-fuzzy system NFS[14].

Tab. 1. Testing sets in texture recognition tasks

Set symbol	Set content description
MT	Includes patterns represented by two best MaZda features (in the sense of F factor evaluation calculated for vector of features).
PFT	Includes patterns represented by two best features selected from the set of all features extracted with PFA.
MST	Includes patterns represented by two best features selected from the set of all features extracted with PFAMS.
CT	Includes patterns represented by two best features selected from the set of all features extracted with PFAC.
MSCT	Includes patterns represented by two best features selected from the set of all features extracted with PFAMSC.
CMST	Includes patterns represented by two best features among all generated with PFACMS.
PF&MT	Includes patterns represented by the best two features selected from the set of all features extracted with MaZda application and PFA.
MS&MT	Includes patterns represented by the best two features selected from the set of all features extracted with MaZda application and PFAMS.
C&MT	Includes patterns represented by the best two features selected from the set of all features extracted with MaZda application and PFAIC.
MSC&MT	Includes patterns represented by the best two features selected from the set of all features extracted with MaZda application and PFAMSC.
CMS&MT	Includes patterns represented by the best two features selected from the set of all features extracted with MaZda application and PFACMS

The applied testing procedure is as follows:

Each method was tested with the use of the leave-one-out algorithm. The k-NN

method was tested for $k = 1, 2, 8$. MLP had one hidden layer containing from 2 to 8 neurons. Fuzzy sets for NFS were defined basing on feature distributions, and the fuzzy rules base was constructed using the table look-up scheme[14].

6.2. Different phases of angisarcoma

The second experiment concerned recognition of different phases of angiosarcoma – a rare but very malignant cancer of blood vessels [6], in the case of which only an early diagnosis can give the patient a chance for recovery. The testing images ([6] p. 211) do not resemble grain textures. Although certain fragments turn out to be reiterative, the repetitiveness is disturbed and irregular. We cut out 5 non overlapping squares of 128 pixel side, from each of the six images. As a result, we obtained a set of 30 images representing six classes. Then we extracted, a total of 675 SFs, 135 with each pseudo-fractal algorithm. Moreover, analogously to the previous one, features were extracted by means of MaZda program.

We have applied the same classifiers as in the previous experiment, i.e. k-NN, NM, MLP, and NFS. Except for the fact that the algorithms were tested with a ten-fold cross validation method, the testing procedure was like the one described earlier.

7. Results

Apart from examinations of classification correctness of SFs we have done an analytical analysis and computer experiments in order to determine computational complexity of pseudo-fractal algorithms. The results are as follows:

The dependence of the number of elementary operations NEO (real number of multiplications and additions) executed by PFA on the number of the image pixels n cannot be expressed in an equation form, as the time is dependent on the type of an image. Therefore, the NEO with respect to n may at most be restricted by inequalities. Analytical analysis of PFA computational complexity resulted in the worse-case performance $O(n^2)$ and the best-case performance $O(n)$. The worse case occurs when for each range from the i hollow every domain from the j hollow is tested. The best case occurs when for each range from the i hollow only one domain is tested. Unfortunately, modifications of PFA have the same the best and worse-case performance. It is obvious that average NOE executed by PFAMS is from 2 up to 16 times less than NOE executed by PFA, but the computational complexity remains the same. Because of the strong dependence of NOE of PFAC, PFAMSC and PFACMS on the image contents, the analysis of average-case performance of the algorithms is extremely difficult and we have not done it yet.

In order to examine the execution time of pseudo-fractal algorithms and compare results with the analysis presented above, we took advantage of angiosarcoma images [6]. We used images with the following spatial resolutions: 64×64, 128×128 and

256×256 pixels. The algorithms were examined under Windows XP with the use of PC with Pentium IV and 2,4 GHz processor and 512 MB of RAM. The results show that PFAMS is form 2 times (for the fifth stage) up to 20 times (for the first stage faster than PFA. Moreover results confirmed the supposition that PFAC execution time is strongly dependant on image contents. In experiments performed PFAC generated features from two to three times faster than PFA. PFAMSC is from 1.5 (for the fifth stage) up to 40 times (for the first stage) faster than PFA. Finally, we observed that feature generation time of PFACMS is from 1.5 (for the fifth stage) up to 12 times (for the first stage) shorter than feature generation time of PFA.

However, example times of feature generation are strong dependent not only on exploited hardware but also on operational system and programs running simultaneously during test, we present some results in order to give general view on the issue. Before running tests we closed all other programs to provide pseudo-fractal algorithms with the greatest possibly computer resources. Finally, for a square image of 128 pixel side, with $Nmax = 5$, $Nmin = 1$, and $AErr = 0$, generation time t for PFA is about 3 minutes. If for the same parameters, the size of the image under examination is 256×256 pixels, t rises up to 24 minutes. By taking into account the type of an image when selecting the algorithm parameter values it is possible to significantly shorten the feature extraction time. Take for instance an image of 128×128 pixels, $Nmax = 5$, $Nmin = 4$, and $AErr = 5$; then $t = 54$ sec. Further acceleration can be achieved by diminishing Nmax; e.g. for an image of 128128 pixels, $Nmax = 3$, $Nmin = 1$, and $AErr = 5$, the extraction time $t = 44$ sec. For small pictures, the time needed for feature extraction is short, for example for an image of 6464 pixels, with $Nmax = 5$, $Nmin = 4$, $AErr = 5$, the algorithm needs 12 sec to finish the task. An increase in the image size results in high growth in the feature extraction time; the image size growth from 128×128 pixels to 256×256 pixels makes the time up to several times longer. The testing results for the first experiment, i.e. percent of correct recognitions for each testing set are collected in Table 2. Each row of the table encloses the best percent of correct recognitions among results for all classifiers. Moreover each row contains the abbreviation of the classifier which gave the best result.

An analysis of the data collected in Table 2 shows that the best percent of correct recognitions (100%) was obtained for the following testing sets: MT, MSCT, PF&MT, MS&MT, MSC&MT, CMS&MT. Worse results were obtained for sets, which enclose only SFs. Moreover 100% of correct recognitions for MT means, that there was not possibility to improve results by application of testing sets which consist of one MaZda feature and one SF: (PF&MT, MS&MT, MSC&MT, CMS&MT).

The testing results for the second experiment are presented in Table 3, which is organized in the same way as Table 2. The best result in Table 3 – 90% of correct recognitions was obtained for MS&MT and PF&MT, which consist of one MaZda feature and one SF. Moreover, like in the first experiment, worse results were obtained for sets,

Tab. 2. Recognition results obtained in the first experiment

testing set	percent of correctly recognised patterns	the best classifier
MT	100	kNN
PFT	97	MLP
MST	97	NFS
CT	94	kNN
MSCT	100	kNN, MLP, NFS
CMST	93	kNN
PF&MT	100	MLP
MS&MT	100	MLP
C&MT	97	MLP
MSC&MT	100	kNN, MLP, NFS
CMS&MT	100	kNN, MLP

Tab. 3. Recognition results obtained in the second experiment

testing set	percent of correctly recognised patterns	the best classifier
MT	83	kNN
PFT	60	kNN
MST	57	kNN
CT	52	kNN
MSCT	60	NFS
CMST	60	kNN
PF&MT	90	MLP
MS&MT	90	MLP
C&MT	83	MLP
MSC&MT	80	kNN
CMS&MT	87	MLP

which enclose only SFs. The result obtained for CT is the worst of all and the result obtained for MSC&MT is the worst of all sets which consist of one MaZda feature and one SF.

8. Conclusions

To sum up, SFs provide a quantitative description of the degree of the self-affinity intensity occurring in the image. The description concerns image parts of various spatial resolutions. Extracted features exhibit independence from affine photometric image transformations, which follows directly from the nature of the affine transformation used in the extraction process. In certain cases, SFs show also independence from geometric image transformations such as scaling, rotation, and shifting. As in the proposed methods no image model is assumed beforehand, applicability of the methods is not limited to any determined types of images. It seems, however, that they can be best applied to images characterized by strong self-affinity, like medical textures, e.g. bone tissue images. The methods can make it possible to discriminate between images representing pathological and normal objects making use of SFs and basing on weakening of self-affinity properties in pathological objects. For example bone microstructure images represent strong self-affinity because bones are build from a lot of similar rods or plates or both which form an interconnected network. Therefore SFs extracted from such images have small values. Osteoporosis damages the network which may weaken the self-affinity of the structure (local decrease of density). This could be identified by extraction of SFs which obtain higher values. The paper presents the recognition results concerning two different types of images, where a considerable number of both typical features extracted from an image and SFs have been examined. We achieved the highest discrimination accuracy for the combination of SFs and typical features, which indicates salient usefulness of SFs.

Comparison of presented pseudo-fractal algorithms leads to a conclusion that PFAMS is the best of all. It can be justified by the fact that testing sets which contained PFAMS features were classified with high percent of correct recognitions and MS&MT got the highest percent of correct recognitions in both experiments. Moreover PFAMS is from 1,5 up to 20 times faster that PFA and is merely slower than PFAMSC - the fastest of all pseudo-fractal algorithms. Therefore we advice application of PFAMSC in that sort of recognition tasks in which the time of feature generation is especially important. Unfortunately one should realise that in some cases the algorithm can give not good enough results. As an example we can quote only 80% of correct recognitions of patterns from MSC&MT in the second experiment.

As we mentioned before there are only several papers, which are relatively close to our work. The closest of them is [21], which presents a new method of fractal coding application to classification of images representing isolated objects (Section 2). The

method was applied to classification of handwritten digits and artificially constructed images of trees. Unfortunately the method is not fully automated. Indeed, some input parameters of the fractal feature generation algorithm were established manually with the use of the trial and error method. In our approach there is full automation of each step from feature generation through feature selection to classification. Moreover in our approach fewer constraints are imposed on affine transformations, which results in more accurate exploration of the image and increases probability of finding similar fragments. Moreover our method could be applied to greyscale images of any content. One of the greatest novelty of our approach is the attempt to make quantitative description of similarity intensity between various parts of an image in various scales. We have not found any paper which presents such an approach. It seems that SFs may find application in texture recognition or in identification tasks involving weakened tissue self-affinity caused by pathological changes, like cancer changes, for example. The features might also be applied to identification of structural changes in bones caused by osteoporosis as well as to various identification tasks covering structural changes in various industrial materials. The experiments carried out allow us to advance a few hypotheses about practical usefulness of the proposed approach but verification of the hypothesis is open matter. In order to do so, the algorithms will be tested on real medical tasks based on a suitable collection of images and the results will be consulted and discussed with medical experts. In this way the practical usefulness of the proposed approach is believed to undergo thorough examination and evaluation.

References

1982
[1] Mandelbrot B.: The fractal geometry of nature. Freeman, San Francisco, CA.

1988
[2] Barnsley M.F., Jacquin A.E.: Application of recurrent iterated function systems to images. T. R. Hsing, eds., Visual Communications and Image Processing, Proc. SPIE 1001, 122-131.
[3] Barnsley M.F., Sloan A.: A better way to compress images, BYTE Mag, 13(1) (1988), 215-223.

1990
[4] Jacquin A.E.: A novel fractal block coding technique for digital images. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 4, 2225-2228.

1991
[5] Tadeusiewicz R., Flasiski M.: Pattern Recognition. PWN, Warsaw, (in Polish).
[6] Trojani M.: A Colour Atlas of Breast Histopatology. J. B. Lippincott Company, Philadelphia.

1992
[7] Fisher Y.: Fractal Image Compression. SIGGRAPH'92 Course Notes, [online], `http://inls.ucsd.edu/~fisher/Fractals/\#papers`.

1993
[8] Barnsley M.F.: Fractals Everywhere, 2nd edition, Academic Press, New York.
[9] Masters T.: Practical Neural Network Recipes in C++. Academic Press, San Diego.

1994
[10] Dubuisson M.P., Dubes R. C.: Efficacy of fractal features in segmentation images of natural textures. Pattern Recogn. Lett. 15, 419-431.

1995

[11] Cheng B., Zhang A., Acharya R., Sibata C. : Using fractal coding to index image content for a digital library. Technical Report 95-05, SUNY, Buffalo, NY.

1996

[12] Martyn T.: Fraktale i obiektowe algorytmy ich wizualizacji. Nakom, Poznan, (in Polish).

1997

[13] Kouzani A.Z., He F., Sammut K. : Optimal fractal coding is NP-hard, Proc. of Data Compression Conference, 261-270.

[14] Rutkowska D., Piliski M., Rutkowski L.: Sieci neuronowe, algorytmy genetyczne i systemy rozmyte. PWN, Warsaw, (in Polish).

1998

[15] Materka A., Strzelecki M. : Texture analysis methods - a Review. COST B11 report, Brussels 1998, [online], `http://www.eletel.p.lodz.pl/cost/publications.html`.

[16] Mitra S.K., Murthy C.A., Kundu M.K. : Technique for fractal image compression using genetic algorithm. IEEE Trans. Image Processing 7(4), 586-593.

1999

[17] Kulikowski J.L. : Methods of computer analysis of textures of histological images. Proc. of Fourth Conf. on Computer technologies in Medicine TIM'99, Silesian University, Dep. of Electronics and Computer Systems, Jaszowiec, Poland, 35-47.

[18] Materka A., Strzelecki M., Lerski R., Schad L. : Feature evaluation of texture test objects for magnetic resonance imaging. Proc. of the 5th Conference on Computers in Medicine, Lodz, Poland, 101-107.

[19] Mikrut Z., Czwartkowski B. : Log-Hough space as input for neural network, Proc. of the 4th Conf. on Neural Networks and Their Applications, Zakopane, Poland, 268-275.

[20] Theodoridis S., Koutroumbas K.: Pattern Recognition. Academic Press, San Diego.

2000

[21] Baldoni M., Baroglio C., Cavagnino D., Bello G.L. : Use of IFS codes for learning 2D isolated-objects classification systems, Computer Vision and image Understanding 77, 371-387.

2001

[22] Mavromatis S., Boi JM., Sequeira J. : Tissue differentiation by using texture analysis, Proc. of the Sixth Portuguese Conference on Biomedical Engineering, Faro, Portugal.

2002

[23] Cichy P.: Texture Analysis - Application to a Selected Class of Biomedical Images. doctoral dissertation, Faculty of Electrotechnology and Electronics, Technical University of Lodz, (in Polish).

[24] Kcki E., Janaszewski M. : Fractals in medical image recognition. Proc. of Int. Conf. on Computer Vision and Graphics, Zakopane, Poland, 387-392.

[25] Tan T., Yan H. : The fractal neighbor distance measure, Pattern Recogn. 35, 1371-1387.

2003

[26] Janaszewski M.: Fractal methods of feature generation, Ph.D. dissertation, AGH University of Science and Technology, Krakow, (in Polish).

2004

[27] Yokoyama T., Sugawara K., Watanabe T. , Similarity-based image retrieval system using partitioned iterated function system codes, Artif Life Robotics 8, 118-122.

2005

[28] Kulikowski J.L., Wierzbicka D. : Choosing serial tests for discrimination of textures in biomedical images. Biocybernetics and Biomedical Engineering, 25(3), 65-77.

[29] Mozaffari S., Faez K., Ziaratban M. : Character representation and recognition using quad tree-based fractal encoding scheme. Proc. Eighth Int. Donf. Document Analysis and Recognition, 2, 819 - 823

2006

[30] Iano Y., Da Silva F.S., Cruz A.L.M.: A fast and efficient hybrid fractal-wavelet image coder. IEEE Trans. Image Processing, 15(1), 98 - 105.

[31] MaZda application help, [online], `http://www.eletel.p.lodz.pl/merchant/mazda/order1\_en.epl`.

[32] Soundararajan E., Cross J.: Fractal-based texture analysis, [online], `http://www.cosc.iup.edu/sezekiel/Poster/Poster1a.doc`.