FRACTAL RECOGNITION OF SHAPES

Krzysztof J. Gdawiec[1]

Abstract: **From the beginning of fractals discovery they found a great number of applications. One of those applications is fractal recognition. In this paper we introduce a fractal recognition method which is based on fractal description obtained from fractal image compression. Next, we present simple modification of this method and results of the tests.**

***Index Terms*:** **fractals, fractal compression, pattern recognition.**

I. INTRODUCTION

Image recognition is one of the most diverse areas of machine vision. The aim of object recognition is to classify unknown images or areas of images, known as objects using known objects. In general all objects in a known class have parameters extracted from them and these parameters are used to classify unknown objects. An ideal image recognition technique would be robust to changes in scale, rotation, illumination effects and noise while being fast to implement. Unfortunately such a technique does not exist.

In the 1970's Benoit Mandelbrot introduced to the world new field of mathematics. He named this field fractal geometry (*fractus* – from Latin divided, fractional). Fractal geometry breaks the way we see everything, it gives us tools to describe many of the natural objects which we cannot describe with help of the classical Euclidean geometry. Fractals existed considerably earlier, but they were perceived as exceptional objects, mathematical monsters. Mandelbrot turned the official interpretation of these objects upside down. He observed that what seemed to be an exception was a rule.

Fractals found wide applications in many domains for example in image compression, generating shore lines, mountains, clouds, plants or in medicine and economy. In this paper we present one of such application – fractal recognition. The fractal recognition methods found application in face recognition [1], [2], [3], [4], signature verification [5], character recognition [6], [7] or as a general recognition method [8], [9].

In the beginning we introduce the notion of fractal and a few basic definitions. Next, we present the idea of fractal image compression which we will use in fractal recognition. We describe a fractal recognition method called Multiple Mapping Vector Accumulator (MMVA) [7]. Next, we present simple modification of this method and results of the tests.

II. FRACTAL AS ATTRACTOR

Because the notion of the fractal can be defined in many ways e.g. as an invariant measure [10], as an attractor [11] or as a set for which Haussdorf dimension is greater than the topological dimension [12] so in this section we will introduce the definition which we will use in this paper. Before we do that we must introduce some notions.

Let us take any complete metric space (X, ρ) and denote as $H(X)$ the space of compact subsets of the X. In this space we introduce the function $h: H(X) \times H(X) \to \mathbf{R}_+$ which is defined as follows

$$h(R,S) = \max\{D(R,S), D(S,R)\}, \qquad (1)$$

where $R, S \in H(X)$ and the mapping $D: H(X) \times H(X) \to \mathbf{R}_+$ is defined as follows

$$D(R,S) = \max_{x \in R} \min_{y \in S} \rho(x,y). \qquad (2)$$

It turns out that the function h is a metric (Haussdorf metric) and the space $H(X)$ is a complete metric space [11]. Another important notion in our considerations is the one of the IFS (Iterated Function System).

We say that a set $W = \{f_1, \dots, f_n\}$, where f_i is contraction mapping for $i = 1, \dots, n$ is an *iterated function system* (IFS). So defined IFS determines the Hutchinson operator which is defined as follows

$$\forall_{A \in H(X)} \; W(A) = \bigcup_{i=1}^{n} f_i(A) = \bigcup_{i=1}^{n} \{f_i(a) : a \in A\}. \quad (3)$$

Let us consider the following recurrent sequence

$$\begin{aligned} &W^0(A) = A \\ &W^k(A) = W(W^{k-1}(A)), k \geq 1 \end{aligned} \qquad (4)$$

where $A \in H(X)$.

The following theorem is a consequence of the Banach fixed point Theorem [11].

Theorem 1. *Let* (X, ρ) *be a complete metric space and* $W = \{f_1, \dots, f_n\}$ *be an IFS. Then exists only one*

[1] kgdawiec@math.us.edu.pl

set $B \in H(X)$ such that $W(B) = B$. Furthermore the sequence defined above is convergent and

$$\underset{A \in H(X)}{\forall} \lim_{k \to \infty} W^k(A) = B. \qquad (5)$$

The limit from the above theorem is called an *attractor* of the IFS or fractal.

III. FRACTAL IMAGE COMPRESSION

To fractal recognition of the objects we need fractal description of them. To find this description we will use fractal image compression [13], [14]. Below we introduce basic algorithm of the fractal compression which we will use later in the process of the recognition.

In the fractal image compression we bring in additional notion of the PIFS (Partitioned Iterated Function System). We say that a set $P = \{(F_1, O_1), \ldots, (F_n, O_n)\}$, where F_i is a contraction mapping, O_i is an area of the image which we transform with $F_i \, (i = 1, \ldots, n)$ is a PIFS. In the case of the image space as the contraction mapping we use affine mappings $F : \mathbf{R}^3 \to \mathbf{R}^3$ of the following form

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ o \end{bmatrix} \qquad (6)$$

where the coefficients $a, b, c, d, e, f \in \mathbf{R}$ describe a geometric transformations and the coefficients $s, o \in \mathbf{R}$ are responsible for the contrast and brightness. In the recognition we restrict to the binary images so in this case the affine transformations take the simplified form

$$F\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix} \qquad (7)$$

The compression algorithm can be described as follows. We divide an image into fixed number of non-overlapping range blocks. We create a list of a domain blocks. The list consist of an overlapping areas of the image, larger than the range blocks (usually two times larger) and transformed using the following mappings

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \qquad (8)$$

These four mappings are transformations of the rectangle (identity, 180° rotation and two symmetries of a rectangle).

Next for every range block R we search for the domain block D such that the value $\rho(R, F(D))$ is the smallest, where ρ is a metric and F is a transformation determined by the position of the R and D blocks, the size of this blocks in relation to oneself and one of the four transformations defined above. This is the most time consuming step of the algorithm.

Fig. 1 presents the idea of the fractal image compression.

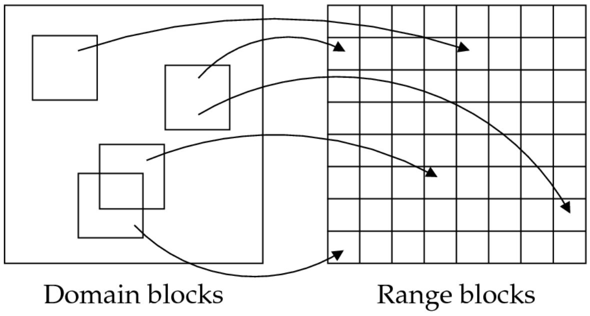

Fig.1 Fractal image compression

IV. FRACTAL RECOGNITION

There exists many methods of recognition of the 2D shapes. In this section we present a fractal recognition method of the 2D shapes using the so-called Multiple Mapping Vector Accumulator [7].

A. *Multiple Mapping Vector Accumulator*

The method which use the Multiple Mapping Vector Accumulator (MMVA) has been presented in [7] and was used to recognition of the handwritten characters. Because we can treat handwritten character as 2D shape we will use this method to the recognition of the 2D shapes. This method consist of the following stages:

- image binarization,
- give the object a correct orientation,
- find normalized PIFS i.e. for which the space is $[0,1]^2$,
- from the PIFS calculate a MMVA vector, we mark it as v,
- recognition in which
  - for each vector from the base v_k calculate $e_k = \rho(v, v_k)$ (ρ – metric),
  - choose an image from the base for which the value e_k is the smallest.

Giving the object a correct orientation is a process in which we rotate the object that it fulfills following conditions: area of the bounding box is the smallest, height of this bounding box is smaller than the width and left half of the object has at least so much pixels as right.

Fig. 2 presents examples of giving the object a correct orientation. In the case of the triangle we see three different orientations. If we add object with several orientations to the base for each of the orientations we find the corresponding PIFS, compute fractal features and these features are added to the base. In the case of recognition of the object with several orientations simply we choose one of the orientations.

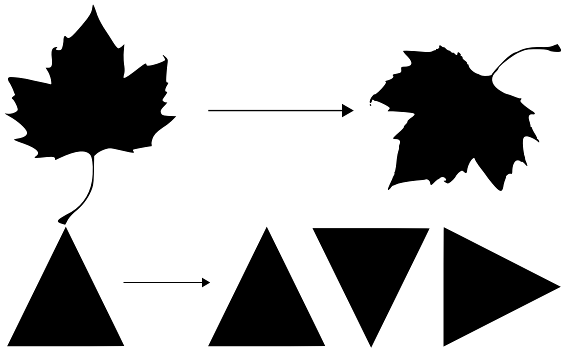

Fig.2 Examples of giving the object a correct orientation

MVA vector contains informations of the relative locations of the range blocks and corresponding domain blocks. Firstly we fix a number of the intervals on which we divide the full angle and the maximum length $M = \sqrt{X^2 + Y^2}$ in the image. Next we create four matrices filled up with zeros. The number of rows in each of these matrices is equal to the number of parts on which we divide the maximum length and the number of columns is equal to the number of parts on which we divide the full angle. For each of the transformation in the PIFS we take the number of transformation used in the mapping of the domain block onto range block and the translation vector. The number of transformation determine the number of the matrix and for the vector we compute the angle and the length of the vector (fig.3). Then, we add 1 into the proper matrix on the location determined by the angle and the length of the vector. For example assume that we want to divide the full angle into 5 parts and $M = 25$ also into 5 parts. Moreover assume that $\alpha = 80°$ and $m = 12$. Following matrix shows the new entry in this case.

	$\le 72°$	$\le 144°$	$\le 216°$	$\le 288°$	$\le 360°$
$\le M/5$					
$\le 2M/5$					
$\le 3M/5$		$+1$			
$\le 4M/5$					
$\le M$					

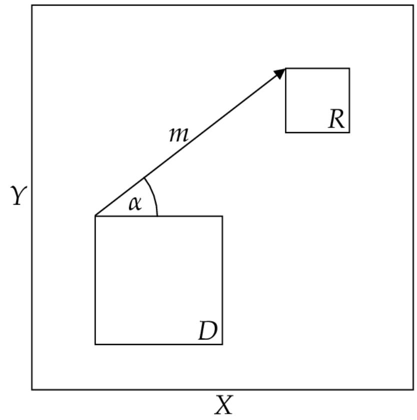

Fig.3 Domain to range vector

After creating the four matrices we reshape them into vectors and then we concatenate these vectors into one MMVA vector.

V. MODIFICATION OF THE MMVA METHOD

Now we introduce a simple modification of the method presented in the previous section. This modification is based on some observation which we will explain with the help of fig.5 and fig.6.

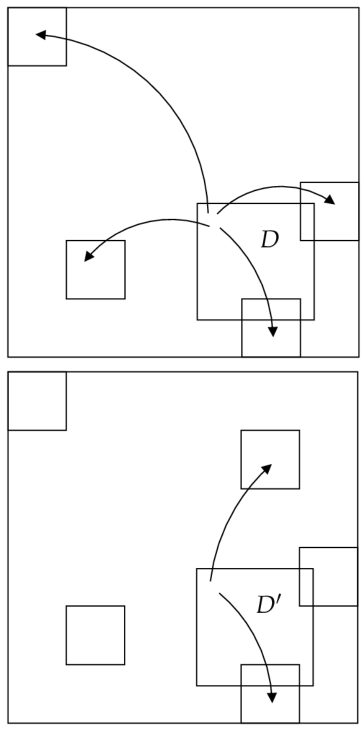

Fig.5 Transformations for the image treated as a whole. Starting situation (top). Situation after the change of block D into D' (bottom)

In fig.5 we can see the situation corresponding to the method presented in the previous section in which one domain block D fits to several range blocks. Now suppose that block D was changed to the block D' (e.g. shape was cut or it was deformed), like in the right figure in fig.5. In this situation block D' doesn't fit to some of the range blocks which fitted to the block D. This cause the change of the transformations of the PIFS.

Now we divide an image into several non-overlapping sub-images e.g. into 4 sub-images (fig.6 top). Again we consider the same domain block D and range blocks. This time block D fits only to the range blocks from the same sub-image, the other range blocks from other sub-images fit to other domain blocks. Now we suppose that block D was changed to the block D' (fig.6 bottom). The change of the block has only influence on the sub-image in which the block D' is placed. Fitting in the other sub-images doesn't change. Therefore locally change of the block D has only local influence on the transformations of the PIFS and not global like in the case of treating the image as a whole.

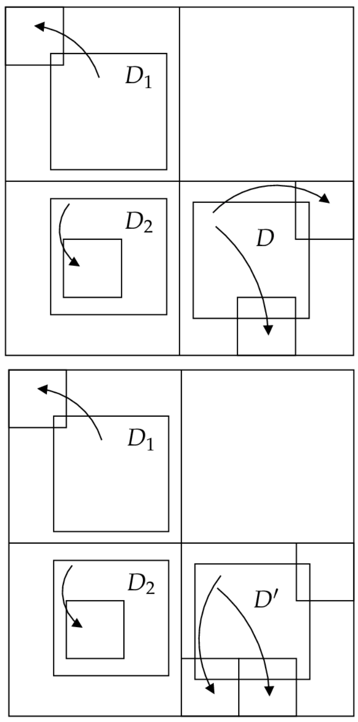

Fig.6 Transformations after subdividing the image into the sub-images. Starting situation (top). Situation after the change of block D into D' (bottom)

The modification of the method is following. In the step with fractal compression we divide image into several sub-images. Next each sub-image is compressed separately. Now the PIFS of the image is a union of the PIFS for the sub-images. Next, we compute the MMVA vectors for each of the sub-images. To measure the similarity between shapes we define new similarity measure as follows:

$$S_{MMVA}(V,U) = \sum_{i=1}^{n} \rho(v_i, u_i) \tag{9}$$

where $V = \{v_1,\dots,v_n\}$, $U = \{u_1,\dots,u_n\}$, n – number of sub-images, $v_1,\dots,v_n,u_1,\dots,u_n$ – MMVA vectors for the sub-images.

VI. RESULTS

Experiments were performed on two databases. The first database was created by the author and the second base was MPEG7 CE-Shape-1 database [15].

Our base consists of three datasets. In each of the dataset we have 5 classes of objects, 10 images per class. The base for creation of the datasets were images from fig.7.

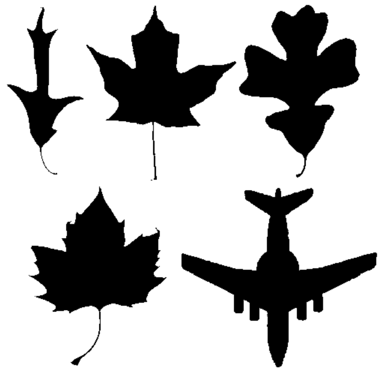

Fig.7 Base images for the datasets

In the first dataset we have base objects changed by elementary transformations (rotation, scaling, translation). Fig.8 presents sample images from the first dataset.

Fig.8 Sample images from the first dataset

In the second dataset we have objects changed by elementary transformations and we add locally small changes of the shape e.g. shapes are cut and/or they have something added. Fig.9 presents sample images from the second dataset.

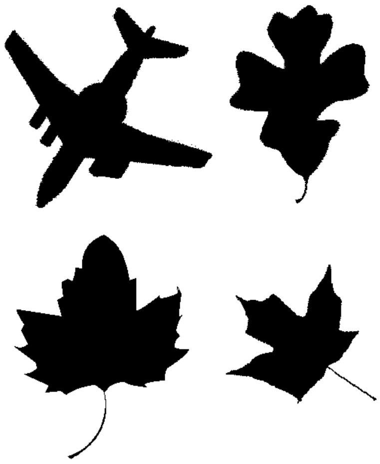

Fig.9 Sample images from the second dataset

In the last, the third set similarly to other two sets the objects were modified by elementary transformations and we add to the shape locally large changes e.g. shapes are cut and/or they have something added. Fig.10 presents sample images from the third dataset.

Fig.10 Sample images from the third dataset

MPEG7 CE-Shape-1 Part B database consists of 1400 silhouette images from 70 classes. Each class has 20 different shapes. Fig.11 presents some sample images from the MPEG7 CE-Shape-1 Part B database.

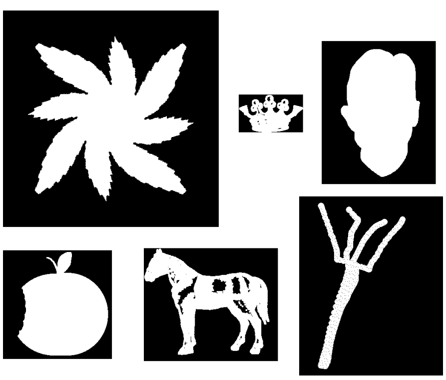

Fig.11 MPEG7 CE-Shape-1 Part B sample images

In the tests we used 256 transformations to the fractal compression. In the modified method we used partition into 4 (2×2) and 16 (4×4) sub-images. Each of the sub-image was compressed by 64 (8×8) transformations for the division into 4 sub-images and 16 (4×4) transformations for the division into 16 sub-images. The size of MVA matrices was 5×5 and as the metric in the recognition process we used the Euclidean distance.

To estimate error rate of the MMVA method and modified method we used leave-one-out method for the three datasets created by the author and for the MPEG7 CE-Shape-1 Part B we used stratified 10-fold cross validation.

Tables 1-3 presents the results of the tests for the base created by the author and table 4 presents the results for the MPEG7 CE-Shape-1 Part B base. In each of the tables the parameters of the method are given in following form: first we give the partition of image into sub-images e.g. 2×2 means that we divide the image into 2 by 2 sub-images, next we give the division of each of the sub-images into range blocks e.g. 8×8 means that we divide the sub-image into 8 by 8 range blocks.

Table 1. Results of the tests for the dataset with elementary transformations

Method parameters	**Error rate [%]**
1×1, 16×16 (original)	18.00
2×2, 8×8	12.00
4×4, 4×4	8.50

Table 2. Results of the tests for the dataset with locally small changes

Method parameters	**Error rate [%]**
1×1, 16×16 (original)	24.00
2×2, 8×8	17.00
4×4, 4×4	12.50

Table 3. Results of the test for the dataset with locally large changes

Method parameters	**Error rate [%]**
1×1, 16×16 (original)	45.00
2×2, 8×8	38.00
4×4, 4×4	32.00

Table 4. Results of the tests for the MPEG7 CE-Shape-1 Part B base

Method parameters	**Error rate [%]**
1×1, 16×16 (original)	49.03
2×2, 8×8	42.01
4×4, 4×4	24.59

VII. CONCLUSIONS

The results of the performed experiments shows that the proposed modification of the MMVA method bring in significant decrease of the recognition error. The error rate become smaller with the division of the image into more sub-images. Moreover we note that speed of the recognition process has also grown up. This speed improvement is caused by the fact that in the case of dividing the image into sub-images and then compressing them the list of the domain blocks on which we are doing the search process are smaller than in the original case.

In our further work we will concentrate on taking into account the number of matching sub-images in the similarity measure, which may bring further fall of the recognition error. Moreover we will perform tests with other divisions of the image into independent sub-images to see the influence of different divisions on the recognition rate. Furthermore we will search for the optimal division into sub-images. Also we will try to bring in our modification into other known fractal recognition methods.

REFERENCES

[1]. A.F. Abate, R. Distasi, M. Nappi, D. Riccio, *Face Authentication Using Speed Fractal Technique*, Image and Vision Computing, vol. 24, pp. 977-986, (2006).

[2]. S. Chandran, S. Kar, *Retrieving Faces by the PIFS Fractal Code*, Workshop on Applications of Computer Vision, pp. 8-12, (2002).

[3]. T. Tan, *Human Face Recognition Based on Fractal Image Coding*, PhD thesis, School of Electrical and Information Engineering, University of Sydney, (2003).

[4]. A. Z. Kouzani, *Classification of Face Images Using Local Iterated Function Systems*, Machine Vision and Applications, vol. 19, no. 4, pp. 223-248, (2008).

[5]. K. Huang, H. Yan, *Signature Verification Using Fractal Transformation*, Proceedings International Conference on Pattern Recognition, vol. 2, pp. 855-858, (2000).

[6]. S. Mozaffari, K. Faez, M. Ziaratban, *Character Representation and Recognition Using Quadtree-based Fractal Encoding Scheme*, IEEE ICDAR, pp. 819-823, (2005).

[7]. S. Mozaffari, K. Faez, F. Faradji, *One Dimensional Fractal Coder for Online Signature Recognition*, ICPR'06, pp. 857-860, (2006).

[8]. G. Neil, K. M. Curtis, *Shape Recognition Using Fractal Geometry*, Pattern Recognition, vol. 30, No. 12, pp. 1957-1969, (1997).

[9]. T. Yokoyama, K. Sugawara, T. Watanabe, *Similarity-based Image Retrieval System Using Partitioned Iterated Function System Codes*, Artifical Life and Robotics 8, pp. 118-122, (2004).

[10]. M. Barnsley, *Superfractals*, Cambridge University Press, (2006).

[11]. M. Barnsley, *Fractals Everywhere*, Academic Press, Boston, (1988).

[12]. B. Mandelbrot, *The Fractal Geometry of Nature*, W.H. Freeman and Company, New York, (1983).

[13]. Y. Fisher, *Fractal Image Compression*, Springer-Verlag, (1995).

[14]. D. Saupe, R. Hamzaoui, H. Hartenstein, *Fractal Image Compression: An Introductory Overview*, Fractal Models for Image Synthesis, Encoding and Analysis, SIGGRAPH '96 Course Notes XX, (1996).

[15]. L. J. Latecki, R. Lakamper, T. Eckhardt, *Shape Descriptors for Non-rigid Shapes with a Single Closed Contour*, Proceedings IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 424-429, (2000).

Krzysztof J. Gdawiec received the M.Sc. degree in Mathematics from University of Silesia in 2005. Currently he is a Ph.D Student in the Institute of Mathematics of the University of Silesia. His research interests include applications of fractal geometry in pattern recognition, image processing and computer graphics.