Shape Recognition Using Partitioned Iterated Function Systems

Krzysztof Gdawiec

Institute of Mathematics, University of Silesia, Bankowa 14, 40-007 Katowice, Poland, kgdawiec@math.us.edu.pl

Summary. One of approaches in pattern recognition is the use of fractal geometry. The property of the self-similarity of the fractals has been used as feature in several pattern recognition methods. In this paper we present a new fractal recognition method which we will use in recognition of 2D shapes. As fractal features we used Partitioned Iterated Function System (PIFS). From the PIFS code we extract mappings vectors and numbers of domain transformations used in fractal image compression. These vectors and numbers are later used as features in the recognition procedure using a normalized similarity measure. The effectiveness of our method is shown on two test databases. The first database was created by the author and the second one is MPEG7 CE-Shape-1PartB database.

1 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.

One of approaches in pattern recognition is the use of fractal geometry. Fractal geometry breaks the way we see everything, it gives us tools to describe many of the natural objects which we cannot describe with the help of classical Euclidean geometry. The property of the self-similarity of fractals was used as feature in several pattern recognition methods. The fractal recognition methods found applications in face recognition [2] [5], signature verification [4], character recognition [7], gait recognition [10] or as a general recognition method [8] [9]. Most of these methods as fractal features use Partitioned Iterated Function Systems (PIFS) and some of them fractal dimension. In this paper we present a new fractal recognition method which we will use in the recognition of 2D shapes. As fractal features we used PIFS. From the PIFS

code we extract mappings vectors and numbers of domain transformations used in fractal image compression. These vectors and numbers are later used as features in the recognition procedure.

First we introduce the notion of a fractal (fractal as attractor [1]) which we will use in this paper and some basic information about fractals. Next, we briefly present fractal image compression [3], which gives us the PIFS code used later in the recognition process. Then, we present our fractal recognition method of 2D shapes which is based on mappings vectors and numbers of domain transformations obtained from PIFS code. The effectiveness of our method is shown on two test databases. First database was created by the author and the second one is MPEG7 CE-Shape-1 Part B database.

2 Fractals

The notion of a fractal differs from many others mathematical notions. It has several nonequivalent definitions e.g. as attractor [1], as an invariant measure [1]. So firstly, we introduce the definition which we will use in this paper. Next, in this section we briefly present fractal image compression method.

2.1 Fractal as Attractor

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

$$h(R, S) = \max\{\max_{x \in R} \min_{y \in S} \rho(x, y), \max_{y \in S} \min_{x \in R} \rho(y, x)\}, \tag{1}$$

where $R, S \in \mathcal{H}(X)$.

The space $\mathcal{H}(X)$ with metric h is a complete metric space [1].

Definition 1. *A transformation $w : X \to X$ on a metric space (X, d) is called a* contraction mapping *if there exists a constant $0 \leq s < 1$ such that for all $x, y \in X$*

$$d(f(x), f(y)) \leq sd(x, y). \tag{2}$$

Any such number s is called a contractivity factor for w.

Definition 2. *We say that a set $W = \{w_1, \ldots, w_N\}$, where w_n is a contraction mapping with contractivity factor s_n for $i = 1, \ldots, N$ is an* iterated function system *(IFS).*

So defined IFS determines the so-called Hutchinson operator which is defined as follows

$$\forall_{A \in \mathcal{H}(X)} W(A) = \bigcup_{n=1}^{N} w_n(A) = \bigcup_{n=1}^{N} \{w_n(a) : a \in A\}. \tag{3}$$

The Hutchinson operator is a contraction mapping with contractivity factor $s = \max\{s_1, \ldots, s_N\}$ [1]. Let us consider the following recurrent sequence

$$\begin{cases} W^0(A) = A \\ W^k(A) = W(W^{k-1}(A)), \quad k \geq 1 \end{cases}, \tag{4}$$

where $A \in \mathcal{H}(X)$.

The next theorem is the consequence of the Banach Fixed Point Theorem [1].

Theorem 1. *Let (X, ρ) be a complete metric space and $W = \{w_1, \ldots, w_n\}$ be an IFS. Then exists only one set $B \in \mathcal{H}(X)$ such that $W(B) = B$. Furthermore, the sequence defined by equation (4) is convergent and for all $A \in \mathcal{H}(X)$ we have $\lim_{k\to\infty} W^k(A) = B$.*

Definition 3. *The limit from theorem 1 is called an* attractor *of the IFS or fractal.*

2.2 Fractal Image Compression

The fractal image compression method uses the fact that every image has a partial self-similarity. So we use additional notion of a Partitioned Iterated Function System [3].

Definition 4. *We say that a set $P = \{(F_1, D_1), \ldots, (F_N, D_N)\}$ is a* Partitioned Iterated Function System *(PIFS), where F_n is a contraction mapping, D_n is an area of an image which we transform with the help of F_n for $n = 1, \ldots, N$.*

In practice, as the contraction mappings we use affine transformations $F : \mathbb{R}^3 \to \mathbb{R}^3$ of the following form

$$F(\begin{bmatrix} x \\ y \\ z \end{bmatrix}) = \begin{bmatrix} a_1 & a_2 & 0 \\ a_4 & a_5 & 0 \\ 0 & 0 & a_7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} a_3 \\ a_6 \\ a_8 \end{bmatrix}, \tag{5}$$

where coefficients $a_1, a_2, a_3, a_4, a_5, a_6 \in \mathbb{R}$ describe a geometric transformation and coefficients $a_7, a_8 \in \mathbb{R}$ are responsible for the contrast and brightness.

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

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{6}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{7}$$

These four mappings are transformations of the rectangle (identity, 180° rotation and two symmetries of the rectangle). Next, for every range block R we look for the domain block D so that the value $\rho(R, F(D))$ is the smallest, where ρ is a metric, F is a transformation determined by the position of R and D, the size of these in relation to itself and one of the four mappings defined by equations (6) (7) and coefficients a_7, a_8 are calculated with the help of equation (8) (9). This is the most time-consuming step of the algorithm.

$$a_7 = \frac{k\sum_{i=1}^{k} g_i h_i - \sum_{i=1}^{k} g_i \sum_{i=1}^{k} h_i}{k\sum_{i=1}^{k} g_i^2 - (\sum_{i=1}^{k} g_i)^2}, \tag{8}$$

$$a_8 = \frac{1}{k}\left[\sum_{i=1}^{k} h_i - s_n \sum_{i=1}^{k} g_i\right], \tag{9}$$

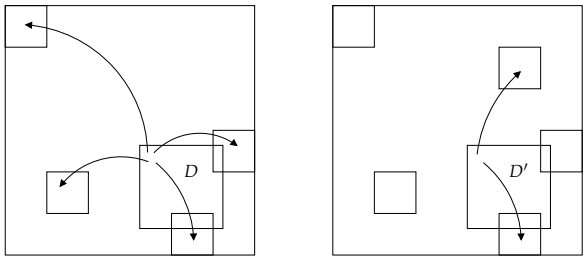
where $g_1, \ldots, g_k$ are the pixel intensities of the transformed and resized domain block, $h_1, \ldots, h_k$ are the pixel intensities of the range block. If $k\sum_{i=1}^{k} g_i^2 - \left(\sum_{i=1}^{k} g_i\right)^2 = 0$, then $a_7 = 0$ and $a_8 = \frac{1}{k}\sum_{i=1}^{k} h_i$.

3 Mapping vectors similarity method

In this section we introduce a new fractal recognition method which is based on the mapping vectors and numbers obtained from transformations of the PIFS. But first, we introduce some observation with the help of which we get a better results of the recognition.

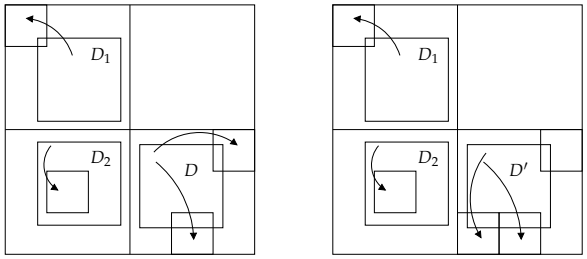
Let us take a look at fractal compression from the point of view of domain block D. Further, assume that this block fits to several range blocks (fig. 1(a)). Each of these fits correspond to one mapping in PIFS. Now let us suppose that block D was changed into block D' (e.g. the shape was cut or it was deformed). This situation is shown in fig. 1(b). In this case domain D' can fit to the same range blocks as D (to all or only to some), it can also fit to some other range blocks. This change of fitting causes a change of the mapping in PIFS. In the worst case all mappings can be changed.

Now we divide the image into several non-overlapping sub-images e.g. into 4 sub-images (fig. 2(a)) and compress each of the sub-images independently. And again let us consider the same domain block D and the same range blocks. This time block D fits only to the range blocks from the same sub-image, the other range blocks from different sub-images fit to other domain blocks D_1, D_2 (fig. 2(a)). Now suppose that block D was changed in the same way as earlier (fig. 2(b)). The change of the block only has an influence on the sub-image in which the block D' is placed. The fitting in other sub-images does not change. This time in the worst case only mappings corresponding to this sub-image change, all the other mappings remain the same. So, locally change of the shape has only a local influence on the transformations of PIFS and not global one as in the previous case.



(a) starting situation (b) situation after the change of D into D'

Fig. 1. Fractal image compression



(a) starting situation (b) situation after the change of D into D'

Fig. 2. Fractal image compression with division

Now we are ready to introduce the mapping vectors similarity method. First, we set the partition of the image into sub-images and the number of mappings of each PIFS. The method looks as follows

1. binarize the image and extract the object,
2. give the object a correct orientation,
3. find a normalized PIFS W i.e. for which the space is $[0,1]^2$, for all sub-images,
4. for each PIFS V from the base
   a) calculate mapping vectors similarity $s = [s_1, \ldots, s_N]^T$ and vector $t = [t_1, \ldots, t_N]^T$,
   b) calculate $e_V = \|\left[\begin{smallmatrix} s \\ t \end{smallmatrix}\right]\|$,
5. choose an object from the base for which the value e_V is the smallest.

Giving the object a correct orientation is a process in which we rotate the object so that it fulfills the following conditions: area of the bounding box is the smallest, height of this box is smaller than the width and left half of the object has at least as many pixels as the right. Figure 3 presents examples of giving an object a correct orientation. In the case of the triangle (fig. 3(b)) we see three different orientations. If we add such an object to the base for

each orientation we find the corresponding PIFS and add it to the base. In the case of recognition we simply choose one of the orientations.

Fig. 3. Examples of giving the object a correct orientation

It remains to tell how to compute the vectors s and t. Firstly, for each of mapping w_n which belong to the PIFS W we take the corresponding mapping v_n from PIFS V. Next, we extract from them the mappings vectors (vector between the range block and corresponding domain block) and mappings used to transform the domain block onto range block (mappings given by equations (6), (7)). Then

$$s_n = 100 - sim(p_n, q_n) = 100 - 50(1 + \cos\alpha)\frac{\min(\|p_n\|, \|q_n\|)}{\max(\|p_n\|, \|q_n\|)}, \tag{10}$$

where p_n and q_n are the mapping vectors for w_n and v_n respectively, and α is the angle between vectors p_n and q_n. Next,

$$t_n = \eta(f_n, g_n) = \begin{cases} 0 & \text{, if } f_n = g_n \\ 1 & \text{, if } f_n \neq g_n \end{cases}, \tag{11}$$

where f_n and g_n are mappings given by equations (6), (7) corresponding to mappings w_n and v_n, respectively.

4 Experiments

Experiments were performed on two databases. The first database was created by the author and the second base was MPEG7 CE-Shape-1 database [6].

Our base consists of three datasets. In each of the datasets, we have 5 classes of objects, 20 images per class. In the first dataset we have base objects changed by elementary transformations (rotation, scaling, translation). In the second dataset we have objects changed by elementary transformations and we add small changes to the shape locally e.g. shapes are cut and/or they have something added. In the last, the third set, similar to the other two sets the objects were modified by elementary transformations and we add to the shape large changes locally.

The MPEG7 CE-Shape-1 Part B database consists of 1400 silhouette images from 70 classes. Each class has 20 different shapes. Figure 4 presents

(a) authors base (b) MPEG7 base

Fig. 4. Sample images

some sample images from the authors base and the MPEG7 CE-Shape Part B base.

In the test we used several different partitions into independent sub-images. One was a partition into 1×1 sub-images which is the classical case. The other partitions were 2×2 and 4×4. The number of transformations used in the fractal compression of each sub-image depends on the partition. For partition 1×1 we used 256 transformations (16×16 range blocks division), for 2×2 – 64 transformations (8×8) and 16 transformations (4×4) for the 4×4 partition.

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

Tables 1(a)-1(c) present the results of the tests for the authors base and tab. 2 presents the results for the MPEG7 CE-Shape-1 Part B base.

Table 1. Results of tests for the authors base

(a) elementary

Partition	Error [%]
1×1	3.00
2×2	2.00
4×4	1.00

(b) locally small

Partition	Error [%]
1×1	6.00
2×2	4.00
4×4	1.00

(c) locally large

Partition	Error [%]
1×1	14.00
2×2	9.00
4×4	8.00

Table 2. Results of tests for the MPEG7 CE-Shape-1 Part B base

Partition	Error [%]
1×1	30.45
2×2	18.58
4×4	16.37

5 Conclusions

A new method for recognition of shapes has been presented in this paper. The method was based on fractal description of the shape. Moreover a modification of the compression scheme was proposed which led to a significant decrease of the recognition error. The division into several independent sub-images also brings an improvement in speed of achieving the fractal description of the shape. This is due to 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 is smaller than in the classical 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 a further decrease of the recognition error. Moreover, we will perform tests with other partitions 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. We will also try to bring the division into sub-images into other known fractal recognition methods.

References

1. Barnsley, M.: Fractals Everywhere. Academic Press, Boston (1988)
2. Chandran, S., Kar, S.: Retrieving Faces by the PIFS Fractal Code. In: Proceedings of the 6th IEEE Workshop on Applications of Computer Vision, pp. 8-12 (2002)
3. Fisher, Y.: Fractal Image Compression: Theory and Application. Springer, New York (1995)
4. Huang, K., Yan, H.: Signature Verification Using Fractal Transformation. In: Proceedings of the International Conference on Pattern Recognition, vol. 2, pp. 855-858 (2000)
5. Kouzani, A.Z.: Classification of Face Images Using Local Iterated Function Systems. Machine Vision and Applications 19(4), 223-248 (2008)
6. Latecki, L.J., Lakamper, R., Eckhardt, T.: Shape Descriptors for Non-rigid Shapes with a Single Closed Contour. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 424-429 (2000)
7. Mozaffari, S., Faez, K., Faradji, F.: One Dimensional Fractal Coder for Online Signature Recognition. In: Proceedings of the International Conference on Pattern Recognition, pp. 857-860 (2006)
8. Neil, G., Curtis, K.M.: Shape Recognition Using Fractal Geometry. Pattern Recognition 30(12), 1957-1969 (1997)
9. Yokoyama, T., Sugawara, K., Watanabe, T.: Similarity-based Image Retrieval System Using Partitioned Iterated Function System Codes. Artifical Life and Robotics 8, 118-122 (2004)
10. Zhao, G., Cui, L., Li, H.: Gait Recognition using Fractal Scale. Pattern Analysis & Applications 10(3), 235-246 (2007)