Title: Pseudofractal 2D shape recognition

Author: Krzysztof Gdawiec

Citation style: Gdawiec Krzysztof. (2010). Pseudofractal 2D shape recognition. "Lecture Notes in Artificial Intelligence" (vol. 6401 (2010), pp. 403-410), doi 10.1007/978-3-642-16248-0_57

[postprint]

Pseudofractal 2D Shape Recognition

Krzysztof Gdawiec

Institute of Computer Science, University of Silesia, Poland
kgdawiec@ux2.math.us.edu.pl

Abstract. From the beginning of fractal discovery they found a great number of applications. One of those applications is fractal recognition. In this paper we present some of the weaknesses of the fractal recognition methods and how to eliminate them using the pseudofractal approach. Moreover we introduce a new recognition method of 2D shapes which uses fractal dependence graph introduced by Domaszewicz and Vaishampayan in 1995. The effectiveness of our approach is shown on two test databases.

1 Introduction

The shape of an object is very important in object recognition. Using the shape of an object for object recognition is a growing trend in computer vision. Good shape descriptors and matching measures are the central issue in these applications. Based on the silhouette of objects a variety of shape descriptors and matching methods have been proposed in the literature.

One of approaches in object 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 idea of fractals was first presented by Mandelbrot in the 1970s [9] and since then have found wide applications not only in object recognition but also in image compression [5], generating terrains [10] or in medicine [7]. Fractal recognition methods are used in face recognition [2], character recognition [11] or as general recognition method [12].

In this paper we present a new approach to the recognition of 2D shapes. In our approach we modified fractal image coding scheme which is base for features in many fractal recognition methods. Moreover we present a new method which uses the so-called fractal dependence graph [3] and our modified coding scheme.

2 Fractals

Because we will modify fractal coding scheme first we must introduce the definition of a fractal. The notion of fractal has several non-equivalent definition, e.g. as attractor or as an invariant measure [1]. The most used definition is fractal as attractor.

Let us take any complete metric space (X, d) and denote as $\mathcal{H}(X)$ the space of nonempty, compact subsets of X. In this space we take the Haussdorf metric:

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

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

Definition 1 ([1]). *We say that a set $W = \{w_1, \ldots, w_N\}$, where w_n is a contraction mapping for $n = 1, \ldots, N$ is an* iterated function system *(IFS).*

Any IFS $W = \{w_1, \ldots, w_N\}$ determines the so-called Hutchinson operator which is defined as follows:

$$\forall_{A \in \mathcal{H}(X)} W(A) = \bigcup_{n=1}^{N} w_n(A) = \bigcup_{n=1}^{N} \{w_n(a) : a \in A\}. \tag{2}$$

The Hutchinson operator is a contraction mapping in space $(\mathcal{H}(X), h)$ [1].

Definition 2. *We say that the limit* $\lim_{k \to \infty} W^k(A)$*, where $A \in \mathcal{H}(X)$ is called an* attractor *of the IFS $W = \{w_1, \ldots, w_N\}$.*

2.1 Fractal Image Coding

Most of the fractal recognition methods use fractal image coding to obtain fractal description of the objects and next extract features from this description. Below we introduce basic algorithm of the fractal coding. Let us start with the notion of partitioned iterated function system (PIFS) [5].

Definition 3. *We say that a set $P = \{(F_1, D_1), \ldots, (F_N, D_N)\}$ is a* partitioned iterated function system*, 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 $n = 1, \ldots, N$.*

Because the definition of PIFS is very general we must restrict to some fixed form for the transformations. In practice we take affine mappings $F : \mathbb{R}^3 \to \mathbb{R}^3$ of the following form:

$$F(\begin{bmatrix} x \\ y \\ z \end{bmatrix}) = \begin{bmatrix} a_1 & a_2 & 0 \\ a_3 & a_4 & 0 \\ 0 & 0 & a_7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \\ a_8 \end{bmatrix}, \tag{3}$$

where coefficients $a_1, \ldots, a_6 \in \mathbb{R}$ describe a geometric transformation, coefficients $a_7, a_8 \in \mathbb{R}$ are responsible for the contrast and brightness and x, y are the co-ordinates in image, z is pixel intensity.

The coding algorithm can be described as follows. We divide an image into a fixed number of non-overlapping areas of the image called range blocks. Next we create a list of domain blocks. The list consists of overlapping areas of the image, larger than the range blocks (usually two times larger) and transformed using four mappings: identity, rotation through 180°, two symmetries of a rectangle.

Now for every range block R we search for a domain block D such the value $d(R, F(D))$ is the smallest, where d is a metric and F is mapping of the form (3) determined by the position of R and D, the size of these in relation to itself, one of the four mappings used to transform D and the coefficients a_7, a_8 are calculated with the help of following equations:

$$a_7 = \frac{k \sum_{i=1}^{k} g_i h_i - \sum_{i=1}^{k} g_i \sum_{i=1}^{k} h_i}{k \sum_{i=1}^{k} g_i^2 - (\sum_{i=1}^{k} g_i)^2}, \quad (4)$$

$$a_8 = \frac{1}{k} \left[\sum_{i=1}^{k} h_i - a_7 \sum_{i=1}^{k} g_i \right], \quad (5)$$

where k is the number of pixels in the range block, $g_1, \ldots, g_k$ 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 $a_8 = \frac{1}{k} \sum_{i=1}^{k} h_i$.

The search process in the most time-consuming step of the algorithm [5]. Figure 1 presents the idea of the fractal image coding.

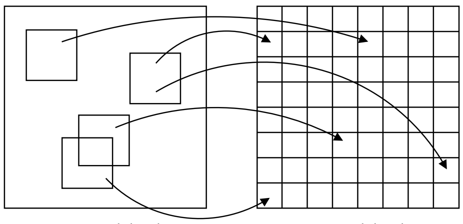


Fig. 1. The idea of fractal image coding

3 Pseudofractal Approach to 2D Shape Recognition

Fractal methods like method using PIFS Code (PIFSC) [2] or Multiple Mapping Vector Accumulator (MMVA) [11] have one weakness. To show this weakness let us take a look at fractal coding from the point of view of some domain block D. This block can fit to several range blocks. Each of these fits corresponds to one mapping in PIFS. Now let us suppose that block D was changed into block D' (e.g. the shape in this area was cut or it was deformed). 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 mappings in PIFS. In the worst case all mappings can be changed. If the change of fitting is big then it can cause that object will be wrongly classified [6].

To eliminate the problem we can use pseudofractal approach. Simply in the fractal coding step for the creation of the domain list we take any fixed image called *domain image*. The rest of the coding algorithm remains the same. Using the domain image the change of fitting can appear only for the ranges where the image was changed and the rest of fitting remains the same because the change of the image has no influence on the domain blocks.

Definition 4. *Fractal image coding using fixed domain image is called a* pseudofractal image coding *and the PIFS obtained with the help of this algorithm is called* pseudofractal PIFS *(PPIFS).*

To obtain better recognition results in the methods mentioned earlier (PIFSC, MMVA) instead of the fractal image coding we can use the pseudofractal image coding. Moreover, before the pseudofractal image coding we can resize the image with the object to fixed dimensions (e.g. 128×128). This step is used to shorten the time needed to coding the image. The PIFSC method modified in the presented way is called pseudofractal PIFSC (PPIFSC) and the MMVA method is called pseudofractal MMVA (PMMVA).

3.1 Pseudofractal Dependence Graph Method

In 1995 Domaszewicz and Vaishampayan introduced the notion of a dependence graph [3]. The graph reflects how domain blocks are assigned to range blocks. They used this graph for three different purposes. The first one is an analysis of the convergence of the fractal compression. The second is a reduction of the decoding time and the last is improvement upon collage coding. The definition of a dependence graph is as follows [3]:

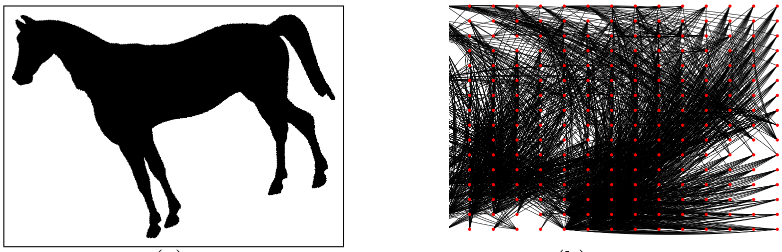
Definition 5. *Let W be the PIFS with a set of range blocks $\mathcal{R}$.* The dependence graph *of W is a directed graph $G = (V, E)$ whose $V = \mathcal{R}$ and for all $R_i, R_j \in \mathcal{R}$ we have $(R_i, R_j) \in E$ if and only if the domain block corresponding to R_j overlaps the R_i.*

Example of an image and a dependence graph corresponding to the PIFS which codes the image are presented in Fig.2.

Definition 6. *The dependence graph for a PPIFS W is called a* pseudofractal dependence graph.

The Pseudofractal Dependence Graph (PDG) method is following:

1. binarise the image and extract the object,
2. find a set of correct orientations Γ,
3. choose any correct orientation $\gamma \in \Gamma$ and rotate the object through γ,
4. resize the image to 128×128 pixels,
5. find a normalised PPIFS W, i.e. for which the space in $[0, 1]^2$,
6. determine the adjacency matrix $\boldsymbol{G}$ of the pseudofractal dependency graph of W,



(a) (b)

Fig. 2. Example image (a) and its fractal dependence graph (b)

7. in the base find adjacency matrix $\boldsymbol{H}$ which minimizes expression

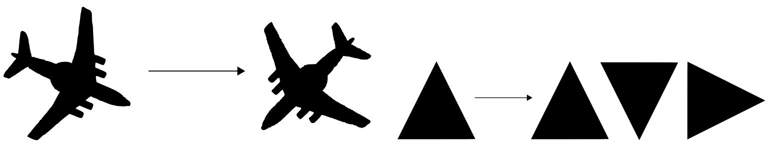
$$d_{\boldsymbol{H}} = \|\boldsymbol{G} - \boldsymbol{H}\| = \sqrt{\sum_i \sum_j |g_{ij} - h_{ij}|^2}, \tag{6}$$

8. choose an image from the base which corresponds to the $d_{\boldsymbol{H}}$.

The notion of a correct orientation used in the PDG method is defined as follows.

Definition 7. *A* correct orientation *is an angle by which we need to rotate an object so that it fulfills following conditions: area of the bounding box is the smallest, height of the bounding box is smaller than the width and left half of the object has at least as many pixels as the right.*

Examples of objects and their correct orientations are presented in Fig.3. In the case of triangle (Fig.3(b)) we see three different orientations. If we want to add such an object to the base for each of the correct orientations, we find the corresponding pseudofractal dependence graph of the PPIFS representing the rotated object and add it to the base.



(a) (b)

Fig. 3. Examples of objects and their correct orientations

Giving the object a correct orientation is needed to make the method rotation invariant. Resizing image to 128 × 128 is used to speed up the fractal coding process and the normalisation is needed to make the method translation and scale invariant.

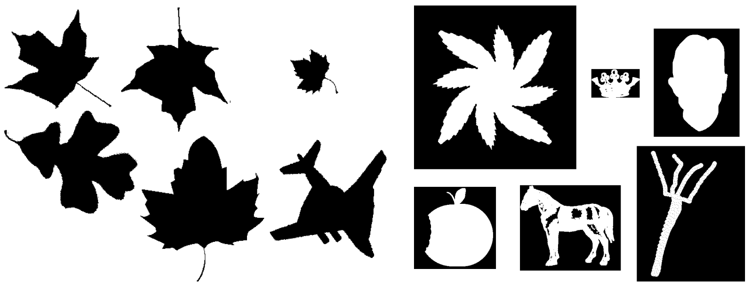
4 Experiments

Experiments were performed on two databases. The first database was created by the authors and the second one was MPEG7 CE-Shape-1 Part B database [8].

Our base consists of three datasets. For creation of the datasets we have used 5 different base images of different classes of objects. In each of the datasets we have 5 classes of objects, 20 images per class. So each dataset consists of 100 images and the whole database of 300 images.

In the first dataset we have base objects changed by elementary transformations (rotation, translation, change of scale). In the second dataset we have base 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 to it. In the last, the third dataset, similar to the other two datasets the objects were modified by elementary transformations and moreover we add to the shape large changes locally. Example images from the authors base are presented in Fig.4(a).

The MPEG7 CE-Shape-1 Part B database consists of 1400 silhouette images from 70 classes. Each class has 20 different shapes. Sample images from the MPEG7 database are presented in Fig.4(b).



(a) authors base (b) MPEG7 database

Fig. 4. Example images form the bases used in the tests

In the tests we used one domain image (Fig.5). The number of transformations used in fractal coding was fixed to 256 transformation (division into 16×16 range blocks).

To estimate the error rate of the methods we used leave-one-out method for the three datasets created by the authors and for the MPEG7 CE-Shape-1 Part B database we used stratified 10-fold cross validation. All the algorithms were implemented in Matlab and the tests were performed on computer with AMD Athlon64 X2 6400+, 4GB DDR2 RAM memory and WindowsXP operating system.

The results of the test for the authors database are presented in Tabs 1(a)-1(c). We can clearly see that the proposed pseudofractal approach led to the

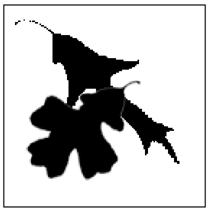


Fig. 5. Domain image used in the tests

decrease of the recognition error for the methods known from the literature. The PDG method has obtained the lowest value of the error on the first and the third dataset. Only on the second dataset the PMMVA method was better, but the difference was small. Moreover we see that the proposed resizing of the image before the fractal coding has shorten the time of the recognition.

Table 1. Results of the test for the authors base.

(a) elementary

Method	Error [%]	Time [s]
PIFSC	4.00	7915
PPIFSC	2.00	1348
MMVA	10.00	7964
PMMVA	2.00	1411
PDG	2.00	1020

(b) locally small

Method	Error [%]	Time [s]
PIFSC	11.00	7123
PPIFSC	4.00	1304
MMVA	18.00	7054
PMMVA	2.00	1364
PDG	3.00	991

(c) locally large

Method	Error [%]	Time [s]
PIFSC	37.00	6772
PPIFSC	4.00	1313
MMVA	32.00	6734
PMMVA	4.00	1368
PDG	3.00	998

The results of the test for the MPEG7 CE-Shape-1 Part B database are presented in Tab.2. Similarly as in the case of the authors base we can see that proposed pseudofractal approach led to the decrease of the recognition error for the methods known from the literature and the PDG method obtained the lowest value of the recognition error.

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

Method	Error [%]
PIFSC	52.74
PPIFSC	18.15
MMVA	49.03
PMMVA	19.22
PDG	14.72

5 Conclusions

In this paper we have presented the weaknesses of the fractal recognition methods and proposed a pseudofractal approach to eliminate the weaknesses. Moreover we have presented a new recognition method called Pseudofractal Dependence Graph Method. The proposed approach led to a significant decrease of the recognition error and PDG method obtained the lowest values of errors. Moreover proposed resizing of the image before fractal coding has shorten the time needed for the recognition. We can shorten the time more by implementing the algorithms in C++ and using the General-Purpose Computation on Graphics Processing Units [4].

Because in the test we used only one domain image in our further research we will use different images as the domain image. With this kind of tests we will check if the results are dependent on the choice of the domain image and which properties the domain image must have. Moreover we will try other classification methods. The correct orientation used to align the object is very simple so there is research under way to find a better method for aligning the object.

References

1. Barnsley, M.: Fractals Everywhere. Academic Press, Boston (1988)
2. Chandran, S., Kar, S.: Retrieving Faces by the PIFS Fractal Code. In: Proceedings 6th IEEE Workshop on Applications of Computer Vision, pp. 8-12 (December 2002)
3. Domaszewicz, J., Vaishampayan, V.A.: Graph-theoretical Analysis of the Fractal Transform. In: Proceedings 1995 International Conference on Acoustics, Speech, and Signal Processing, vol. 4, pp. 2559-2562 (May 1995)
4. Erra, U.: Toward Real Time Fractal Image Compression Using Graphics Hardware. In: Bebis, G., Boyle, R., Koracin, D., Parvin, B. (eds.) ISVC 2005. LNCS, vol. 3804, pp. 723-728. Springer, Heidelberg (2005)
5. Fisher, Y.: Fractal Image Compression: Theory and Application. Springer, New York (1995)
6. Gdawiec, K.: Shape Recognition using Partitioned Iterated Function Systems. Advances in Intelligent and Soft Computing 59, 451-458 (2009)
7. Iftekharuddin, K.M., Jia, W., Marsh, R.: Fractal Analysis of Tumor in Brain MR Images. Machine Vision and Applications 13, pp. 352-362 (2003)
8. Latecki, L.J., Lakamper, R., Eckhardt, T.: Shape Descriptors for Non-rigid Shapes with a Single Closed Contour. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 424-429 (June 2000)
9. Mandelbrot, B.: The Fractal Geometry of Nature. W.H. Freeman and Company, New York (1983)
10. Meng, D., Cai, X., Su, Z., Li, J.: Photorealistic Terrain Generation Method Based on Fractal Geometry Theory and Procedural Texture. In: 2nd IEEE International Conference on Computer Science and Information Technology, pp. 341-344 (2009)
11. Mozaffari, S., Faez, K., Faradji, F.: One Dimensional Fractal Coder for Online Signature Recognition. In: Proceedings of the 18th International Conference on Pattern Recognition, vol. 2, pp. 857-860 (August 2006)
12. Neil, G., Curtis, K.M.: Shape Recognition Using Fractal Geometry. Pattern Recognition 30(12), 1957-1969 (1997)