Recognition of Two-dimensional Shapes Based on Dependence Vectors

Krzysztof Gdawiec and Diana Domańska

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

Abstract. The aim of this paper is to present a new method of two-dimensional shape recognition. The method is based on dependence vectors which are fractal features extracted from the partitioned iterated function system. The dependence vectors show the dependency between range blocks used in the fractal compression. The effectiveness of our method is shown on four test databases. The first database was created by the authors and the other ones are: MPEG7 CE-Shape-1PartB, Kimia-99, Kimia-216. Obtained results have shown that the proposed method is better than the other fractal recognition methods of two-dimensional shapes.

1 Introduction

Nowadays recognition of objects is very important task. Because of that the research on methods of recognition is very intensive and most diverse area of machine vision. Often object is represented by its shape so good shape descriptors and matching measures are the central issue in the research. The shape descriptors can be divided into two groups: based on the silhouette and based on the contour of the object. In both of the groups there are many known methods [1].

From the beginning fractals gain much attention. First in the computer graphics, because the images of fractals were perceived as very interesting and beautiful. Later fractals found applications in other areas of our life, e.g. in economics, medicine, image compression [2]. With the help of fractals we were able to represent real world objects much better than with the help of the classical Euclidean geometry, so this was the motivation to use fractal as a shape descriptor for the recognition. The methods based on fractal description of the shape found applications in: face recognition [3], character recognition [9], general recognition method [10], etc.

In the paper we present a new method which is based on fractal features. The features are called dependence vectors and are extracted from the partitioned iterated function system, which is obtained from the fractal compression of the image containing the object.

In Section 2 we briefly introduce the definition of a fractal and the fractal compression scheme, which will be later used in our method. Next, in Section 3 we present the notion of dependence vectors and a method of recognizing two-dimensional shapes. Later, in Section 4 we give the description of conducted experiments and used test bases. Finally, in Section 5 we give some conclusions.

2 Fractals and Fractal Image Compression

In this section we will present the fractal image compression scheme used later in the proposed recognition method. But first we need to introduce the notion of a fractal, because there are several non-equivalent definitions. The definition we use in this work is fractal as attractor [2].

First we must define the notion of an iterated function system (IFS) [2].

Definition 1. *Let (X, d) be a metric space. We say that a set of mappings $W = \{w_1, \ldots, w_N\}$, where $w_n : X \to X$ is a contraction mapping for $i = 1, \ldots, N$ is an* iterated function system*.*

Any IFS $W = \{w_1, \ldots, w_N\}$ determines the so-called Hutchinson operator which is contrative mapping on the space $(\mathcal{H}(X), h)$, where $\mathcal{H}(X)$ is the space of non-empty, compact subsets of X and h is the Haussdorf distance [2]. The Hutchinson operator is given by following formula:

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

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

The fractals poses the property of self-similarity, i.e. any part of the fractal is similar to the whole fractal. The real world objects do not have this property. Instead they have partial self-similarity, i.e. smaller parts of object are similar to bigger parts of the object [7]. The fractal image compression is based on the partial self-similarity and the notion of a partitioned iterated function system (PIFS).

Definition 3. *We say that a set $P = \{(F_1, A_1), \ldots, (F_N, A_N)\}$ is a* partitioned iterated function system*, where F_n is a contraction mapping and A_n is an area of the image which is transformed with the help of F_n for $n = 1, \ldots, N$.*

In practice as the mappings from the Definition 3 we use affine mappings of the space $\mathbb{R}^3$ of the following form:

$$F(\begin{bmatrix} x \\ y \\ z \end{bmatrix}) = \begin{bmatrix} a_1 & a_2 & 0 \\ a_3 & a_4 & 0 \\ 0 & 0 & a_7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \\ a_8 \end{bmatrix}, \tag{2}$$

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.

In the coding scheme given later we have two types of blocks: range and domain. The set of range blocks consists of non-overlapping blocks of the same size that cover the image. The number of range blocks is fixed before we start the coding. The set of domain blocks consists of overlapping blocks bigger than the range blocks (usually two times bigger) and transformed using four mappings: identity, rotation through 180°, two symmetries of a rectangle.

The fractal coding scheme is following:

1. Create a set of range blocks $\mathcal{R}$ and domain blocks $\mathcal{D}$.
2. For each range block $R \in \mathcal{R}$ find domain block $D \in \mathcal{D}$ such that

$$D = \arg\min_{D' \in \mathcal{D}} \rho(R, F(D')), \tag{3}$$

where ρ is a metric (usually Euclidean), F is a mapping of the form (2) determined by the position of R and D', the size of the blocks in relation to itself, one of the four mappings used to transform D' and the coefficients a_7, a_8 are calculated by following formulas:

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

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

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 $a_8 = \frac{1}{k} \sum_{i=1}^{k} h_i$.
3. Remember the coefficients of F and block D.

The search process in step 2 is the most time-consuming step of the coding algorithm [7].

This algorithm is very simple and therefore used only in fractal image recognition. Moreover, in recognition of two-dimensional shapes in binary images the coefficients a_7 and a_8 are omitted. In practice, when we compress an image we use adaptive methods of partitioning such as quad-tree, HV partition and others [7].

3 Dependence Vectors Method

In our previous works [4] [5] [6] we have shown some weaknesses of the fractal recognition methods and how to improve them. In [5] [6] we proposed division of the image into sub-images and compression of each sub-image independently. Better improvement was obtained in [4] using the pseudofractal approach in which we use fixed image as the source for domain blocks in the fractal compression algorithm. The pseudofractal approach will be used in the proposed method which we call dependence vectors method (DVM).

Before we give the dependence vectors method we need to introduce the definition of the dependence vectors.

Definition 4. *Let W be the PIFS with a set o range blocks $\mathcal{R}$. For each $R \in \mathcal{R}$ we define* dependence vectors *of R as a set of vectors between the range block R and the range blocks that overlap the domain block corresponding to R. Set $V = \{V^1, \ldots, V^N\}$, where V^i are dependence vectors of R_i for $i = 1, \ldots, |\mathcal{R}|$ is called* set of dependence vectors.

Figure 1 presents one range blocks, corresponding domain block (bold line), range blocks that overlap the domain block (dashed grey line) and the dependence vectors.

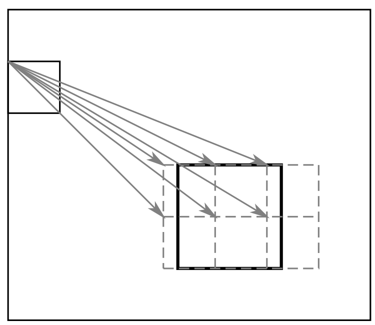

Fig. 1. Range block and its dependence vectors.

The DVM method is following:

1. extract object from the image,
2. find a set o correct orientations $\varGamma$,
3. choose a correct orientation $\gamma \in \varGamma$ and rotate the object through γ,
4. resize the image to 128×128 pixels,
5. find normalized PIFS W using the pseudofractal approach,
6. determine the set of dependence vectors V_W of W,
7. in the base $\mathcal{B}$ find a set of dependence vectors V such that

$$V = \arg\min_{V_B \in \mathcal{B}} \sum_{i=1}^{N} h(V_B^i, V_W^i), \quad (6)$$

   where N is the number of range blocks, h is the Haussdorf distance based on the Euclidean distance,
8. choose an image from the base which corresponds to V.

Some of the points in the method need further explanations. First one is the notion of correct orientation. A correct orientation is an angle by which we need to rotate an object so that it fulfils following conditions: area of the bounding box is the smallest, height of the bounding box is smaller than the width and the left half of the object has at least as many pixels as the right. The correct orientation is needed because we want the method to be rotation invariant.

Resizing the image to 128×128 pixels is used to speed up the fractal coding process and the normalization is needed to make the method translation and scale invariant.

4 Experiments

To show effectiveness of the proposed method we compare it with other existing fractal methods. These methods are: Neil-Curtis method (NC) [10], Multiple Mapping Vector Accumulator (MMVA) [9], method which uses the PIFS coefficients (CM) [3], Mapping Vectors Similarity Method (MVSM) [5], Fractal Dependence Graph Method (FDGM) [6]. All the methods will be tested in the original form. Moreover all the methods, except the Neil-Curtis method, will be tested using the pseudofractal approach presented by the authors in [4] and in this case the abbreviations of the methods will begin with the letter P, e.g. PMMVA for the pseudofractal MMVA.

In the test we used division into 16×16 range blocks, so PIFS consists of 256 transformations. As the source for domain blocks in the pseudofractal approach we used one image showed in Fig. 2.

The description of the databases used in the tests and the obtained results are shown in the next subsections.

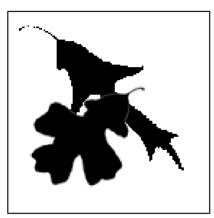

Fig. 2. Source for the domain blocks used in the tests.

4.1 Authors Base

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, i.e. 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. Finally, in 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 large changes are made in such a way that the shape is still recognizable.

In the tests on our three datasets to estimate the error rate we used leave-one-out method. The results of the tests are shown in Tabs. 1(a)-1(c). From the Tab. 1(a) we see that the lowest values of error (2%) obtained almost all methods so we can say that they are robust to elementary transformations. For the base with locally small changes (Tab. 1(b)) we see that the best result (2%) was obtained by the PPMVA method. The proposed method with other two methods (PMVSM, PFDGM) obtained error rate equal to 3%. Finally, from the Tab. 1(c) we see that the proposed method with the PFDGM method obtained the best result (3%). Two other methods: PMMVA and PCM obtained error rate equal to 4% and the rest of the methods obtained error rate greater than 10%.

Table 1. Results of the test for the authors base

(a) elementary

Method	Error [%]
DVM	2.0
NC	2.0
MMVA	10.0
PMMVA	2.0
CM	4.0
PCM	2.0
MVSM	3.0
PMVSM	2.0
FDGM	2.0
PFDGM	2.0

(b) locally small

Method	Error [%]
DVM	3.0
NC	4.0
MMVA	18.0
PMMVA	2.0
CM	11.0
PCM	4.0
MVSM	6.0
PMVSM	3.0
FDGM	6.0
PFDGM	3.0

(c) locally large

Method	Error [%]
DVM	3.0
NC	11.0
MMVA	32.0
PMMVA	4.0
CM	37.0
PCM	4.0
MVSM	14.0
PMVSM	10.0
FDGM	14.0
PFDGM	3.0

4.2 MPEG7 CE-Shape-1 Part B Database

The MPEG7 CE-Shape-1 Part B database [8] consists of 1400 silhouette images from 70 classes. Each class has 20 different shapes.

For the estimation of the error rate we used the stratified 10-fold cross validation. The obtained results are shown in Tab. 2. From the results we see that the proposed method obtained the lowest value of the error (13.15%). Moreover we see that the pseudofractal versions of the methods have values of the error very close to the error of the DVM method.

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

Method	Error [%]
DVM	13.15
NC	29.45
MMVA	49.03
PMMVA	19.22
CM	52.74
PCM	18.15
MVSM	30.45
PMVSM	17.58
FDGM	35.88
PFDGM	14.72

4.3 Kimia Databases

The Kimia-99 database [11] consists of object images from 9 different classes. In each class we have 11 shapes. The last database used in the test is Kimia-216 [11]. The base consist of 216 images selected from the MPEG7 CE-Shape-1 Part B. The images are divided into 18 classes, 12 images per class.

Like in the case of authors base for the Kimia databases we used the leave-one-out method for the estimation of the error rate. The obtained results for the Kimia-99 are shown in Tab. 3(a) and for Kimia-216 in Tab. 3(b). From the obtained results we clearly see that the DVM method obtained the best results (12.12% for the Kimia-99, 13.42% for the Kimia-216). Similarly like for the MPEG7 base only the pseudofractal versions of the methods obtained results close to the best result.

Table 3. Results of the test for the Kimia bases

(a) Kimia-99

Method	Error [%]
DVM	12.12
NC	15.15
MMVA	45.45
PMMVA	17.17
CM	43.43
PCM	14.14
MVSM	32.32
PMVSM	22.22
FDGM	29.29
PFDGM	13.13

(b) Kimia-216

Method	Error [%]
DVM	13.42
NC	14.81
MMVA	33.79
PMMVA	14.81
CM	31.48
PCM	14.81
MVSM	27.31
PMVSM	17.12
FDGM	26.38
PFDGM	14.35

5 Conclusions

In the paper we have presented a new method of two-dimensional shape recognition, which we called Dependence Vectors Method. In the method as the features we used dependence vectors and also we used the psuedofractal approach proposed by the authors in [4]. The experiments have shown that the proposed method obtained smaller error rates comparing to the other known fractal recognition methods.

In our further work we will concentrate on improving the effectiveness of our method with the help of using different types of classifiers, other similarity measures. Moreover we will conduct further research to check if the pseudofractal approach depends on the image chosen for the creation of the domain set. All the tested methods used descriptors from the whole shape, so we will try to find contour descriptor which is based on fractal description.

References

1. Amit, Y.: 2D Object Detection and Recognition: Models, Algorithms, and Networks. MIT Press, Cambridge (2002)
2. Barnsley, M.: Fractals Everywhere. Academic Press, Boston (1988)
3. 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)
4. Gdawiec, K.: Pseudofractal 2D Shape Recognition. In: Yu, J., Greco, S., Lingras, P., Wang, G., Skowron, A. (eds.) RSKT 2010. LNCS (LNAI), vol. 6401, pp. 403-410. Springer, Heidelberg (2010)
5. Gdawiec, K.: Shape Recognition using Partitioned Iterated Function Systems. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stanczyk, U., Wakulicz-Deja, W. (eds.) Man-Machine Interactions. AISC, vol. 59, pp. 451-458. Springer, Heidelberg (2009)
6. Gdawiec, K., Domańska, D.: Partitioned Iterated Function Systems with Division and Fractal Dependence Graph in Recognition of 2D Shapes. International Journal of Applied Mathematics and Computer Science 21(4), 757-767 (2011)
7. Fisher, Y.: Fractal Image Compression: Theory and Application. Springer, New York (1995)
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. 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)
10. Neil, G., Curtis, K.M.: Shape Recognition Using Fractal Geometry. Pattern Recognition 30(12), 1957-1969 (1997)
11. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of Shapes by Editing Their Shock Graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5), 550-571 (2004)