2011 International Conference on Document Analysis and Recognition

Online handwriting recognition of Tamil script using Fractal geometry

Rituraj Kunwar, A. G. Ramakrishnan
MILE Lab, Department of Electrical Engineering,
Indian Institute of Science (IISc),
Bangalore, India.
rituraj@mile.ee.iisc.ernet.in, ramkiag@ee.iisc.ernet.in

***Abstract*—We present a fractal coding method to recognize online handwritten Tamil characters and propose a novel technique to increase the efficiency in terms of time while coding and decoding. This technique exploits the redundancy in data, thereby achieving better compression and usage of lesser memory. It also reduces the encoding time and causes little distortion during reconstruction. Experiments have been conducted to use these fractal codes to classify the online handwritten Tamil characters from the IWFHR 2006 competition dataset. In one approach, we use fractal coding and decoding process. A recognition accuracy of 90% has been achieved by using DTW for distortion evaluation during classification and encoding processes as compared to 78% using nearest neighbor classifier. In other experiments, we use the fractal code, fractal dimensions and features derived from fractal codes as features in separate classifiers. While the fractal code is successful as a feature, the other two features are not able to capture the wide within-class variations.**

***Keywords*-Fractal geometry; Online handwritten character recognition; Online handwriting recognition; Fractal coding; Tamil character recognition; Tamil handwriting recognition**

I. Introduction

Fractal objects generally have complex fine structures when viewed at arbitrarily small scale. These objects are too crinkled to be described by conventional geometrical measures like length, area or volume; instead they are often characterized by fractal dimension which is strictly greater than the topological dimension. They could be considered as redundant objects as they are either made of transformed copies of either themselves or part of themselves. The above facts have been exploited in compression of any random pattern. Fractal codes are the compressed representation of the pattern, generated using fractal geometry.

A comprehensive study has been conducted in the present research to examine the effectiveness of self similarity in encoding 1-D [3] [4] [5] [8] ordered online handwritten Tamil characters. The mathematical theory behind the encoding technique is of iterative contractive transformations in metric spaces based on the work of Barnsley. A simplified version of the fractal block coding technique for digital image [1] [2] [9] has been used to encode the 1-D handwritten patterns. The efficacy of the resultant fractal codes has been tested on recognition of online handwritten characters. A novel adaptive partitioning (where trace is segmented into equal or unequal range segment based on cumulative angle) algorithm has been proposed to reduce the extremely high computation during the encoding and decoding process, with minor fall in recognition accuracy. A significant jump in recognition accuracy has been achieved by using DTW matching for classification.

II. Fractal overview

A new geometry was proposed by Benoit Mandelbrot [10] which explains the complete geometrical property of natural objects around us. This is not possible for classical geometry which defines geometry of objects using approximation towards ideal objects. The difference between natural and idealized objects is that natural objects are rarely differentiable at any point, whereas idealized objects could be. According to fractal geometry, most of the natural curves are of infinite length yet enclosing finite area. Therefore, a completely different form of representation called dimension was introduced.

The dimension of embedding space which is decided by the degree of freedom is not equal to the dimension of objects until and unless the object is an ideal object. The dimension of object depends on how it fills the embedding space; therefore it is less than or equal to the dimension of embedding space. One among the several ways of measuring dimension is box counting method as defined below.

$$Dim = \lim_{\epsilon \to 0} \frac{\ln N(\varepsilon)}{\ln(\frac{1}{\varepsilon})} \tag{1}$$

where: N(ε) = no. of squares required to cover the image (2-d case) or no. of segments joining 2 points (1-d case) and ε = side length of square (2-d case) or length between 2 points (for 1-d case). The evaluated dimension signifies the identity of the object, which represents the crookedness, roughness, smoothness of the object. Fractals are geometrical objects having fractional dimension.

Convergent sequence: A sequence could be of points, images, any random set. One special kind of sequence is "convergent sequence" which satisfies the Cauchy convergent theorem and therefore called Cauchy sequence.

For the current study, we work in a space which is complete, compact and closed. The elements of this space are all possible online handwritten patterns. Here we deal with Cauchy sequences present in this space. Cauchy sequence

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.279

has a limit point at which this sequence converges. This limit point itself is a part of the metric space defined by the property of completeness. So we need to define some type of step by which we can start from any online pattern and on continuously taking that step we converge to a particular pattern. The simplest way of taking that step would be linear mapping. This can be written as:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} a & b \\ l & m \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad (2)$$

Here, a, b, l, m perform rotation and flipping of points x_n and y_n whereas e and f translate the flipped and rotated points by e along x and f along y direction. The parameters a, b, l, m, e and f define the affine transform W. A collection of transformations can be used and we can take the union of all of them to get the final transformed pattern. Iterations of these steps will converge to a particular pattern which is the limit point of this Cauchy sequence. To accomplish this, we need a contractive transformation (W), which satisfies *Banach's contraction mapping theorem*. Transformation is contractive if:

$$\begin{vmatrix} a & b \\ l & m \end{vmatrix} < 1 \quad (3)$$

Banach's Contraction Mapping Theorem: Let (X,d) be a complete (means that all Cauchy sequences in this metric space converge to points which are members of this metric space) metric space (it is the space where the d is defined) and W(X) $\rightarrow$ X is contraction mapping i.e

$$d(W(x), W(y)) < \lambda(d(x, y)),\ 0 < \lambda < 1,\ all x, y \varepsilon X \quad (4)$$

If the above condition is satisfied, then it states that W has a unique fixed point. In other words, it states that if W is applied iteratively on (x, y), then it leads to a Cauchy sequence, which converges to a unique fixed point.

The above concept is used in fractal compression [1], which works on the assumption that there is lot of self similarity in any natural object/pattern. Fractal codes are generated which is the compressed representation of the pattern.

In this work, fractal coding of handwritten characters is carried out segment wise (where each segment contains some fixed length of the character), where the self similarity for each sub segment R_i (range segment in the written character) is found across the written character and appropriate affine transform (W_i) is generated and stored. This collection of affine transforms corresponding to all the sub segments of the pattern is enough to regenerate the concerned original pattern with minor distortion.

III. Building of Fractal Codes for Handwritten Characters

The raw online handwritten character is preprocessed by smoothing, re-sampling the trace into 200 equispaced points and normalizing the bounding box. The preprocessed locus is divided into non-overlapping range segments, each having a fixed number of points (R). Last point of each range is also the first point of the next range.

A. *Creating pool of domain segments:*

The domain pool is formed for each character locus, as the collection of all possible domain segments. The number of points in each domain segment can be anything more than range segment (D $>$ R). In our experiments, D = 2R. A D-point window is first located at the beginning of the stroke. Domain pool is obtained by sliding the window along the stroke, δ points at a time, in such a way that it does not cross the end point of the stroke. δ is chosen as R/2 in our experiments.

B. *Constructing transformed Domain pool:*

Transformed domain pool is constructed by multiplying each of the domain segments with the eight isometrics. Steps are as follows:

1) Each domain is translated to its centroid and scaled down by the contractivity factor (s=0.5).
2) The following transformations are applied to each candidate domain segment.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

The above transformations reflect and rotate the domain segment about different axes, producing a family of geometrically related transformed domain segments.

In domain pool, matching blocks will be looked for, in order to encode the online handwritten character.

C. *Searching most identical domain segment for each range:*

Each of the affine transformed domain segments is re-sampled into R points and then centroid of each of the re-sampled domain is translated to that of the concerned range segment. Distance between them is found. Similarly distances w.r.t to the entire transformed domain is calculated. The most similar domain segment corresponding to each range segment is identified and fractal code is stored corresponding to the particular range.

Fractal codes corresponding to each range segment consists of:

1) The range segment index.
2) The range segment centroid.
3) Index of the most similar domain segment.
4) The domain segment transformation index, i.e. out of 8 transformations, the index of one used is stored.

The above steps are repeated for all the character samples to obtain the fractal codes.

D. Issue related to constructing fractal codes:

The whole character is divided into equisized range segments. Smaller the number of points in each range segment, more minutely the complexity in any region of the character can be captured. Relations between range size, number of range segments and the speed of encoding are given by,

1) number of range segments per character is inversely propotional to number of points in each range.
2) encoding speed per character is inversely propotional to number of range segments per character

There are regions in a character, where the curliness is minimal; in those areas, the size of the range segment could be increased, still encoding the region precisely.

E. Steps to encode a handwritten character where the number of points in each range is variable:

Cumulative angle θ_C is calculated starting from the first point and traversing the character stroke till it crosses a threshold θ_T which is empirically set. Smaller the threshold, finer is the encoding. As shown in the Fig 2.

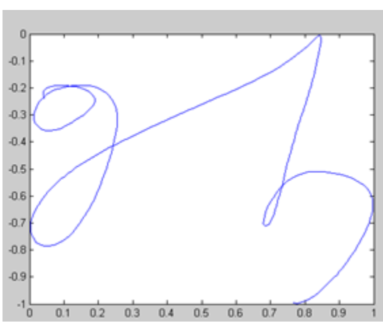

Figure 1. Original Tamil handwritten character /aa/

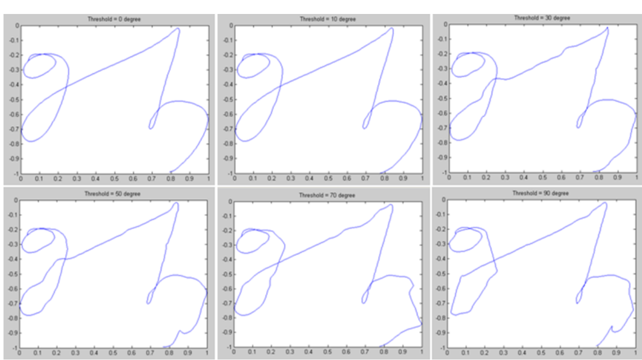

Figure 2. The above figure shows the distortion caused at the end of reconstruction for different choices of the value of the threshold, namely 0, 10, 30, 50, 70 and 90 degrees. Here reconstruction is performed using the fractal codes of the Tamil handwritten character /aa/ (shown in Fig 1). For encoding, different range sizes of 4, 8, 12, 16 and 20 are used.

F. Algorithm for adaptive partitioning

1) Domain pools of different sizes, namely 8, 16, 24, 32 and 40 are constructed corresponding to range sizes of 4, 8, 12, 16 and 20. By size, we mean number of points in each domain.
2) Start from the first point and move along the character from one point to the next and calculate the cumulative change in angle θ_C.
3) The point at which the cumulative angle θ_C crosses the threshold θ_T (i.e. $\theta_C > \theta_T$) is found. The number of points (K) till the penultimate point is noted.
4) The range size which is closest and less than K is chosen. Then the most suitable domain is chosen from the corresponding domain pool and the fractal codes are stored.
5) Then the last point of the present range is considered as the first point of the new range and the process repeats starting from step 2.

- In this case, along with the usual information in the fractal codes, the size of the range chosen is also stored.
- In case the end point is reached with $\theta_C < \theta_T$, then step 4 is followed, with K including the last point also since $\theta_C < \theta_T$).
- If at the end, the left over points are less than the size of the smallest range, then they are discarded.

IV. Reconstruction of a pattern from its fractal codes

Algorithm for reconstruction: Banach's contractive mapping theorem states that if a contractive mapping 'W' (which are the fractal codes here) is defined, then on applying the mapping iteratively on any sequence of the same space, a Cauchy's sequence results which will converge to a unique fixed point.

A. Case I: Range having fixed number of points

1) A random initial pattern having the same number of points is taken or generated.
2) A domain pool of size double that of the range is created in the manner similar to how it was done during the encoding process.
3) First fractal code is taken corresponding to the first range of the pattern, and operations are performed on the corresponding domain indicated by the domain index in code of first range.
4) The indicated domain segment's origin is shifted to its centroid and then it is scaled down by the contractivity factor (in this research it is 0.5).
5) Then the affine transformation as indicated in the code is applied on the scaled domain segment.
6) Finally the transformed domain's centroid is shifted to the range segment centroid as present in the fractal code.

7) The above steps 3 to 6 are repeated to decode the entire range segment.
8) Then the whole decoded locus is smoothened.

The above 3 to 8 steps are repeated till the termination condition is satisfied to finally converge to a fixed and unique pattern.

Termination condition could be:

1) Fixed number of iterations which could be empirically set (enough for a pattern to converge)
2) Minimal or no distortion between the patterns produced by 2 consecutive iterations.

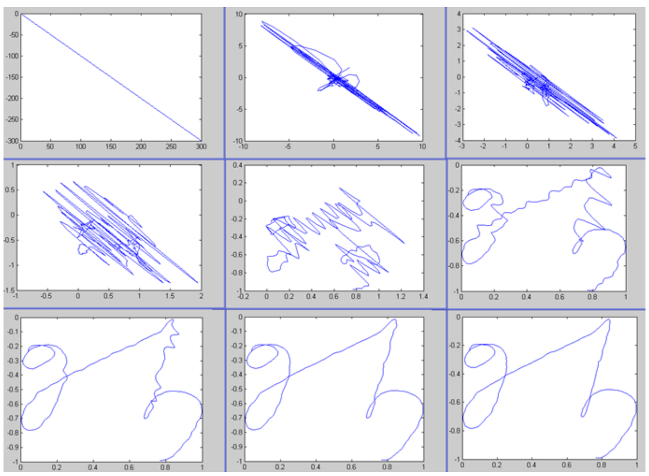

Figure 3. In the above image, reconstruction process is shown which starts from a random straight line and finally converges to a pattern which is very close to the original pattern after 8 iterations. The original pattern (in Fig 1) was encoded using variable range (range sizes of 4, 8, 12, 16 and 20) with a threshold angle of 30 degree.

B. Case II: Range having variable number of points

In this case, multiple domain pools are created out of the random pattern taken for reconstruction. Using the range size information in the fractal code, the domain segment is picked up from the appropriate domain pool. The rest of the steps are the same as the case of reconstruction with fixed range size.

Using above two methods, fractal codes of any given pattern can be created and the same pattern could be decoded using any random pattern after applying this reconstruction algorithm iteratively for a few times, as shown in Fig 3.

V. Character Classification Using Fractal Codes

1. Classification using fractal Codes in construction and reconstruction: The above fractal encoding and decoding method has been used for classification of characters [6] [7]. Assume that a sample of a class (say /aa/) is encoded and fractal codes are obtained. Let the reconstruction process be iteratively applied first on a random pattern of any class other than /aa/ (see Fig. 4) and secondly on any sample of the same class (i.e. /aa/, see Fig. 5). Then the distortion between the initial pattern and the pattern reconstructed after first iteration is higher in the first case than in the second. The reason behind this is that reconstruction process converges to the pattern whose code is used for reconstruction. And since the class of the initial pattern in the second case and the fractal code is same, the distortion in the second case is smaller than the first case.

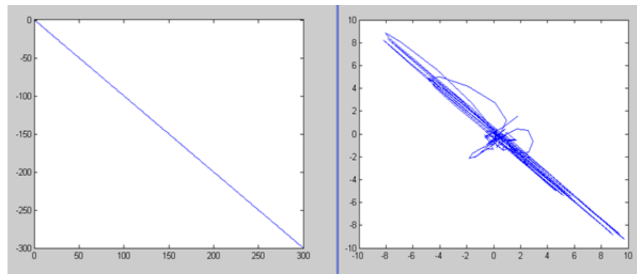

Figure 4. The above image shows the distortion created, when an iteration of reconstruction was performed. This shows that the distortion is huge if the starting pattern (above left) is very different from the original pattern, whose fractal codes are used for reconstruction (in this case Fig 1).

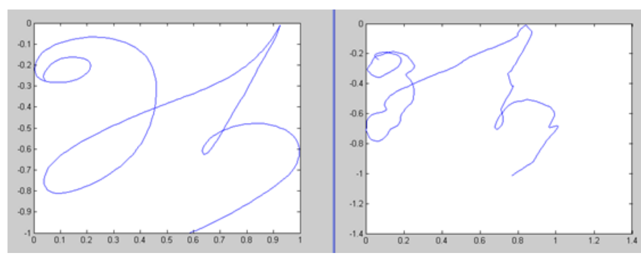

Figure 5. The above image shows the distortion created, when an iteration of reconstruction was performed. This shows that the distortion is much less if the starting pattern (above left)is not very different from the original pattern, whose fractal codes are used for reconstruction (in this case Fig 1)

Classification Algorithm:

In the present research, fractal codes of 'N' samples of all the 156 classes are computed and stored. The test sample is classified as follows.

a) An iteration of the reconstruction algorithm is applied on the test sample using the fractal code of each sample of each class.

b) The distortion 'D' is calculated between the initial pattern and the pattern obtained after one step of reconstruction. Thus the distortion matrix of size 156*N is obtained.

c) The row number (in the matrix) of the minimum value of the distortion is assigned to the test sample.

Distortion evaluation: Using Euclidean distance metric matches the patterns point by point, which can increase the distance unusually. This explains the decrease in classification accuracy seen in Table I. This drawback is addressed by DTW, which matches similar subsections of the patterns

Table I

RESULTS DEMONSTRATING INCREASED ACCURACY WITH DTW.

Fixed Range Size	DTW used?	No. of Training samples	No. of Test samples	Accuracy (%)
4	No	20	50	78.9
4	Yes	20	50	90.4

Table II

EFFICACY OF ADAPTIVE PARTITIONING IN REDUCING ENCODING TIME, WITH MARGINAL DROP IN ACCURACY. RANGE SIZES USED: 4, 8, 16, 24, 32. NO. OF TRAINING AND TESTING SAMPLES USED: 5 AND 30. NO. OF CLASSES:156. CLASSIFIER: DTW.

Threshold angle (degree)	Encoding time per sample (sec)	Accuracy (in %)
0	31	89.0
10	22	86.4
30	12	85.0
50	10	83.6
70	8	81.6
90	6	80.9

thus producing a reasonable distance between them. Thus DTW results in an increased accuracy as shown by Table I.

2. Classification using the fractal codes as features in the Nearest Neighbor: This method gives an accuracy of approximately 65%

3. Classification using fractal dimension or features derived from the fractal codes: Fractal dimension is a unique identity of any pattern or object but because of the mere nature of handwritten character recognition (i.e. large variation within every class), it fails completely to classify any random sample.

Due to the same reason, features derived from the fractal codes like multiple mapping vector accumulator (MMVA) [8] and domain range collocation matrix (DRCLM) [8] completely fail in the case of handwritten character recognition, even though they have been successful in problems like face recognition [9] and signature verification [3] [8].

VI. RESULTS

The classifier was evaluated on IWFHR 2006 competition dataset. The results in Table I show the improvement in accuracy achieved by using DTW during encoding and classification. Table II shows the improvement achieved in encoding time by using adaptive partitioning algorithm. It also shows its effect on the recognition accuracy, which shows that even though fine details of the patterns are marginally lost, the basic characteristics of data are still contained even when the adaptive partitioning algorithm is applied for encoding at as high an angle as 90 degree. Table III compares the recognition accuracy obtained by different classifiers. This shows the ability of fractal method to achieve a good accuracy of 90% with less training, due to the matching done at the small segment level.

Table III

RESULT COMPARISON OF DIFFERENT RECOGNITION METHODS.

Classifier	No. of training samples	No. of Test samples	Recognition time (sec/sample)	Accuracy (%)
Fractal	20	50	135	90.4
HMM	250	100	2.0	85.2
SDTW	250	100	0.5	83.9
SVM	250	180	0.8	86.0

VII. CONCLUSION

The experiments conducted show the potency of adaptive partitioning to decrease the encoding time and efficacy of DTW matching at the time of encoding and classification to improve recognition accuracy to 90.4%. To our best knowledge, this is the maiden report on reducing the fractal encoding time of online data such as handwriting.

ACKNOWLEDGMENT

We thank Mr. Shashikiran for his help. Special thanks to Technology Development for Indian Languages (TDIL), DIT for funding this research.

REFERENCES

[1] Arnaud E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Trans. on Image Processing, 1(1):18-30,1992.

[2] G. Neil and K.M. Curtis, Scale and rotationaly invariant object recognition using fractal transformations, IEEE ICASSP, 6:3458-3461,1996.

[3] Kai Huang and Hong Yan, Signature verification using fractal transformation, IEEE ICPR, 2:851-854,2000.

[4] Mozaffari S., Faez K. and Kanan H.R, Feature comparison between fractal codes and wavelet transform in handwritten alphanumeric recognition using SVM classifier, IEEE ICPR, 2:331-334,2004.

[5] Mozaffari S., Faez K. and Ziaratban M., Character representation and recognition using quad tree-based fractal encoding scheme, IEEE ICPR, 2:851-854,2000.

[6] Rituraj Kunwar, Mohan P, Shashikiran K and A G Ramakrishnan, Unrestricted Kannada Online Handwritten Akshara Recognition using SDTW, Proc International Conf signal processing and communication, IEEE SPCOM, 2010.

[7] Shashikiran K, Prasad K.S., Rituraj Kunwar and A. G. Ramakrishnan, Comparison of HMM and SDTW for Tamil handwritten character recognition, IEEE SPCOM, 2010.

[8] Mozaffari S., Faez K. and Faradji F, One Dimensional Fractal Coder for Online Signature Recognition, IEEE ICPR, 2:857-860, 2006.

[9] T. Tan and H. Yan, Face recognition by fractal transformations, IEEE ICASSP, 6:3405-3408, 1999.

[10] M. F. Barnsley, Fractals everywhere, New York: Academic, 1988.