International Journal of Modern Trends in Engineering and Research
www.ijmter.com
e-ISSN No.:2349-9745, Date: 2-4 July, 2015

PIFS: A New Tool for Face Recognition

Mr. Rakesh S. Shirsath[1], Prof. L. B. Randive[2], Prof. B. S. Ahirwadkar[3]
[1] *ME(CSE), MIT, Aurangabad, rakesh.shirsath@gmail.com*
[2] *Asst. Prof. (CSE), MIT, Aurangabad, l.randive@gmail.com*
[3] *Asst. Prof. (CSE), MIT, Aurangabad, bhaktipr@gmail.com*

Abstract - Biometrics refers to identifying an individual based on his/her physiological or behavioral characteristics. Biometrics is inherently more reliable and accurate than the knowledge-based and token-based systems in differentiating an authenticated person from an imposter. This is based on the fact that physiological or behavioral characteristics are unique to a person. Also, the person to be identified has to be physically present at the time of verification or identification. Biometrics has the potential to replace all the needs of person verification in this electronically interconnected society in near future. Basically, biometric system is a pattern recognition system which recognizes a person using one or more specific characteristics. In this paper, we describe an approach to image coding based on a fractal theory of partitioned iterated contractive transformations. Local contractive affine transformations are iteratively applied to range-domain pairs in an arbitrary initial image results in a fixed point, closed to a given image and is referred as attractor. The main characteristics of this approach are that: i) it relies on the assumption that image redundancy can be efficiently captured and exploited through piecewise transformations on blocks of same image, and ii) it matches an original image approximately by a transformed image, obtained from a finite number of iterations of an image transformation called a PIFS code. PIFS code consists of the transformation parameters like range block number, index of approximately matched domain block, contrast scaling factor and luminance shift. PIFS code are used directly as feature of face. Features of an unknown face image are compared with those precomputed for images in a database.

Keywords - Image coding system, Face recognition; Fractals; Fractal coding; PIFS code.

I. INTRODUCTION

The human face is a very rich source of information that can be used for identification purpose. This ability of recognition allows us to distinguish persons despite the facial resemblance between them. Nowadays, many researchers try to get a benefit from computer applications like face automatic recognition. Face recognition systems which are existing can be classified into following three parts.

A. Local Approach

This is based on the fact that the face contains parts that have a high discriminating power such as eyes, nose, mouth, *etc*. To recognize a person, we use the blocks of these regions or the geometric relationships between them. This work includes hidden Markov model [1], elastic bunch graph matching algorithm [2]. Image recognition method is also based on the self-similarity of blocks of same image which uses the fractal neighbor distance as a parameter for recognition [3].

B. Global Approach

This approach consider face as a whole object and use all the information included in it. Many methods have been developed that include the use of Eigenfaces [4], discrete cosine transforms, and Gabor

Wavelets [5]. These methods suffer from two points: i) the size of the feature vector provided to the classifier and ii) addition of new face in input database.

C. Hybrid Approach

Theme of these approaches is to imitate the human visual system, which based on use of local and global features to recognize faces. Purpose of these two methods is to take advantage of the combined benefits of both approaches [6].

Factors like pose, illumination variations, presence and absence of structural components can affect a performance of face recognition. To ensure a high recognition rate in less time, we have used the fractal representation called PIFS, which exploits the inter-image resemblance[7]. There are few articles that are related to this topic (face recognition using Iterated Function System (IFS) theory [8]).

There are applications of fractal theory of iterated function system in several areas of image processing and computer vision. In this method, similarity between different parts of an image is used for representing of an image by a set of contractive transforms on the images, for which the fixed point is close to the original image called as an attractor. This concept was first proposed by *Barnsley* [9]. *Jacquin* was the first to publish an implementation of fractal image coding in [10]. In this paper we use partitioned iterated function system which is image coding using fractals and quadtree partitioning as described in [11]. Here Face recognition refers to the automatic identification of a person, represented in a database, on the basis of an information in a digital gray scale image. Many methods have been developed for this applications such as principle component analysis (PCA)[12], feature recognition using Neural Networks [13].*Neil et al* introduced an application of fractal transformations in shape recognition and object recognition for use on binary images [14]. *Kouzani et al* used the same method in combination with neural networks [15] and fractal dimension [16] for application of face recognition. After this face recognition using fractal neighbor distance (FND) [3] have been reported. In this paper, we will present a new tool of face recognition which works on the basis of fractal coding on partitioned blocks.The paper is organized as follows: Section II depicts the theoretical foundation, which is required to understand the system. Section III explains the technique of partitioned iterated function system (PIFS) and how it is different from iterated function system. In section IV, we describe a designing of partitioned iterated function system with method of block comparison and transformation operations. Section V illustrates the construction of PIFS block codes for recognition of faces and section VI explains about the experimentation and results in this area followed by conclusion.

II. THEORETICAL FOUNDATIONS

This section provides basic notation and definitions related to PIFS on blocks of images.

A. Metric Space

A space M is a metric space if for any two elements x and y, there exists a real number $d(x , y)$ called distance, that satisfies the following properties:

(1) $d(x , y) >= 0$ (non-negativity)
(2) $d(x , y) = 0$ if and only if x = y (Identity)
(3) $d(x , y) = d(y , x)$ (Symmetry)
(4) $d(x ,z) <= d(x, y) + d(y , z)$ (Triangle inequality)

B. Contractive Mapping

Let $W : X \rightarrow X$ be a transformation on the metric space (X, d) and is said to be contractive with contractivity factor s if for any two points x, y, the distance

$$d(w(x)\,,\,w(y)) < s\,.\,d(x\,,y). \tag{2.1}$$

This formula says the application of a contractive map always brings points close together by some factor less than one. Contractive transformations have the property that when they are repeatedly applied, they converge to a point which remains fixed upon further iteration.

C. Affine Transformation

For a gray scale image *I*, if *z* denotes the pixel intensity at the position *(x, y)*, then affine transformation *W* can be expressed in matrix form as follows:

$$W\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e \\ f \\ o \end{pmatrix} \tag{2.2}$$

This transformation also can be shown in linear form $W(X) = AX + B$, where A is $n\ X\ n$ matrix and B is an offset vector of size $1\ X\ n$. Using an affine transformation, we can perform isometric operations on an image like scaling, rotation, contrast scaling or translation on pixel intensities.

D. Iterated Function Systems(IFS)

As fractals are having a property of self-similarity so their shape is unchanged even applying different isometrics operations. Let *I* be a given image which belongs to the collection of compact sets. Our goal is to find out a set of affine contractive maps such that its fixed point will be a close approximation of

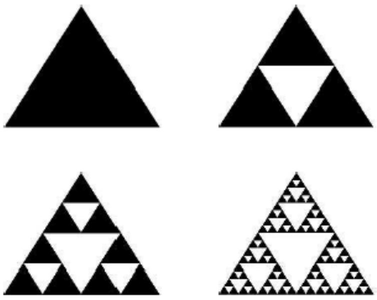

Fig 1: Serpinski Triangle

the given image *I*.On applying sufficient number of iterations of affine contractive transformations, it produces a set which is very close to the given original image *I*. This is called the iterative function system. Fig. 1 shows an example of attractor of an IFS with three simple contractive transformation.

III. PARTITIONED ITERATED FUNCTION SYSTEM METHODOLOGY

The structure of PIFS code is almost same as that of IFS codes. Partitioned iterated function system (also called local iterative function system) is an extension of IFS theory for coding the gray level images. From the application domain point of view the theory of PIFS is different from that of iterated function system. The only difference is that PIFS codes are calculated and applied to a particular portions of the image which are self-similar to each other instead of the whole image [17].

A. What is PIFS?

Partitioned Iterated Function System is a more general type of transformation which works on the the fact that a part of an image can be approximated by a transformed version of another part of the same image.

Let I be a given digital image having size $W \times W$. Thus the given image I can be expressed as a matrix $(g(i, j))_{W \times W}$, where *i* and *j* indicates row number and column number respectively and *g(i, j)* represents the grey level value for the position *(i, j).*

The image is partitioned into *n* non overlapping blocks of size *b X b*, and let this partitions be represented by $r_n = \{r_1, r_2, \ldots r_n\}$. Each r_i is named as range block. The number of range blocks in an image of size $W \times W$ is $n = \left(\frac{w}{b}\right) \times \left(\frac{w}{b}\right)$.

Similarly an image is also partitioned into larger blocks D_j called domain blocks which can be overlapped. Let *M* be a collection of all possible blocks of size *2b X 2b* within the image. Let $M = \{D_1, D_2, \ldots, D_m\}$. Here $m = (W - 2b + 1)(W - 2b + 1)$

Purpose of transformation is to find a domain block of the same image for every range block such that a transformed version of this block $W(D_R)$ is a good approximation of R_i. *W* is a combination of a general transformation and luminance transformation.

Thus PIFS is nothing but the set of parameters of transformation along with contrast and luminance parameters. Distance *dw* is a metric to compute the distance between two PIFS codes by summing-up the term wise absolute differences between the parameters of the two codes [10]. The affine contractive transformation *W* is constructed in such way that the gray values of the range blocks are scaled and shifted version of the gray values of corresponding domain blocks. The contractive affine transformation *W* defined such that $W(D_{R_i}) \rightarrow R_i$.

Here *W* consists of two parts i) for spatial information and ii) for information of gray values of the block. The first part shows pixels of the range block corresponds to pixels of domain block. The first part shuffle the pixel values of the domain block and can achieve transformed block by using any one of the eight possible transformations (isometrics) on the domain blocks [10]. The second part is to find the scaling and shift parameters for the set of pixels of the domain blocks to the range blocks [17]. Once the first part is obtained second part is estimation of a set of values (gray values) of range blocks from the set of values of the transformed domain blocks. The second part is obtained using least square analysis of two sequences of gray values one from the range block and other from the domain block.

Isometric operations can be performed on the domain block to approximate the range block. The luminance part consists a few simple functions, such as a luminance shift and contrast scaling with contrast factor is less than one.

B. How PIFS technique differs from IFS?

An extension of the iterated function system is the partitioned iterated function system. The only difference is in terms of the domain of application of their respective transformations [17]. In PIFS, the transformations are not applied to the whole image as in the case of IFS. In PIFS, a transformation *W* is specified not only by an affine map but also by the domain to which *W* is applied.

IV. DESIGN OF PIFS SYSTEM

Steps involved in the design of a PIFS block coding system are:

i) Partitioning of an image,

ii) Measure of distortion between blocks of two images that is image comparison and

iii) Specification of a finite class of contractive image transformations defined consistently with a partition and of a scheme for the quantization of their parameters [10].

A. Partitioning

To make use of self-similarity property of portions of images (in range and domain blocks), the decision is to be made in the choice of image partitioning techniques.
The simplest possible range partition method consists of the fixed size square blocks. Quadtree partitioning employs the well known image partitioning technique based on recursive splitting of selected image portions, so that the resultant partitions or blocks be represented by a tree structure in which each non-terminal node has four descendants.
A horizontal-vertical (HV) partition produces a tree like structured partition of the image. Here each image block is split into two by a horizontal or vertical line and a finally number of different constructions of a triangular partitions have been investigated.

B. Image Comparison

It is necessary to have a measure of "distance" between images or blocks of images for comparing how closely two blocks match and for using the Contractive Mapping Fixed-Point Theorem. We use the root mean square (rms) metric to measure the similarity between two images (or portions of an image)[18]. If there are blocks of range and domain blocks as $r_s=[r_1, r_2, ... r_n]$ and $d_s=[d_1, d_2, .. d_n]$ then the rms distance between them is defined by

$$basic\ rms = \sqrt{\sum_{i=1}^{n} (d_i - r_i)^2} \qquad (4.1)$$

If d_i represents a domain block which is to have contrast and brightness adjustments c and b respectively, then the distance between the adjusted block and the range block is,

$$adjusted\ rms = \sqrt{\sum ((c \square d_i + b - r_i))^2} \qquad (4.2)$$

The values of c and b that minimize this value, will provide the best match between the blocks r_s and d_s This minimum error occurs when the partial derivatives with respect to c and b are zero, which is possible when following holds:

$$c = \frac{n\sum_{i=1}^{n} d_i . r_i - \sum_{i=1}^{n} d_i \sum_{i=1}^{n} r_i}{n\sum_{i=1}^{n} d_i^2 - \left(\sum_{i=1}^{n} d_i\right)^2} \qquad (4.3)$$

$$b = \frac{1}{n}\sum r_i - c\sum d_i \qquad (4.4)$$

Here c is called contrast scaling factor and b is called luminance shift (that is brightness adjustments). The rms distance between the pixels is then calculated on the basis of values of contrast scaling and luminance shift parameters. These calculated values can be used to find the domain block that best matches each range block, as well as the contrast and brightness adjustments for that block.

C. Class of Discrete Image Transformations

In this section, we describe the class of discrete contractive affine image transformations defined blockwise. The operation of W to transform image blocks from a domain blocks D to a range blocks R can be decomposed into the following stages: Isometric operations, geometric contraction of domain block, parameters like contrast scaling and brightness shift.

4.1 Isometric Operations

The rotation and flip operations do not modify pixel values, they simply shuffle pixels within a block in a deterministic way and we call them isometrics. There are many isometrics operations. The following eight are commonly used operations [7].
i. identity (no rotation or flip operation),
ii. orthogonal reflection about mid-vertical axis,
iii. orthogonal reflection about mid-horizontal axis,
iv. orthogonal reflection about first diagonal of block,
v. orthogonal reflection about second diagonal,
vi. rotation around center of block through +90◦,
vii. rotation around center of block through +180◦,
viii. rotation around center of block through −90◦.
These operations generates from a single block, a whole family of geometrically related transformed blocks, which provides a pool in which matching blocks will be sought during the encoding.

4.2 Geometric Contraction

The domain blocks must be contracted spatially to the size of the range block. In the case when the domain block is twice the size of the range block, the pixel values of the contracted domain block is the average values of four neighboring pixels in the domain block[7].

4.3 Brightness Shift

The brightness shift must be found to make equal brightness of the domain block and the range block. It is nothing but the difference between average pixel values in the range block and average pixel values in the domain block[7].

4.5 Contrast Scaling

A contrast scaling factor is computed to match the contrast among pixels in the domain block and contrast among pixels in the range block. It is a greatest brightness difference among pixels in the range block divided by the greatest brightness difference among pixels in the domain block [7].

V. CONSTRUCTION OF PIFS CODE TO RECOGNIZE FACE

In this section, we describe the step-by-step construction of a PIFS code for any given original image as:

A. Overview of the Procedure

An original *r* x *r* digital image *I*, quantized to *256* gray levels, is given as input. Image *I* is partitioned into non-overlapping range blocks using quadtree partitioning. Let R_i denote the image partitions made of range blocks. Then the same image is partitioned into a sequence of possibly overlapping domain blocks. Let D_j denote the image partitions made of domain blocks. To keep the transformation contractive, the size of a domain block is always larger than the range block so the scale factor is always less than one [17]. For each range block, we need to find the domain block and transformations that best match the range block. Save the geometrical positions of range block and matched domain block and the matching transformation parameters as PIFS code of images.

B. Algorithm

1. Input a binary image as *I*.
2. Cover *I* with non-overlapping range blocks. All range blocks must cover *I* without overlapping.
3. Introduce the domain blocks *D*. The size of the domain block is twice the size of the range block.
4. Define a collection of local contractive affine transformations mappings on domain block D_j to the range block R_i .

5. For each range block, choose a corresponding domain block and symmetry so that the domain block approximates the given range block.
6. Store all the parameters in the form of a PIFS code.
7. Repeat steps 1 to 6 for all references images and query image.
8. Compare the PIFS code of query image with stored code of reference images.

C. Mapping Domains to Range blocks

The main computational step in the algorithm is the mapping of domain to range blocks. For each range block, the algorithm compares transformed version of domain block with the range block as shown in fig.2. The transformation *W* is a combination of a geometrical transformation and luminance transformation. For each range block, the algorithm compares transformed versions of all the domain blocks and eight orientations of each domain block to the selected range block. The transformations are affine and the orientations consist of four 90^0 rotations and the reflected version of each.

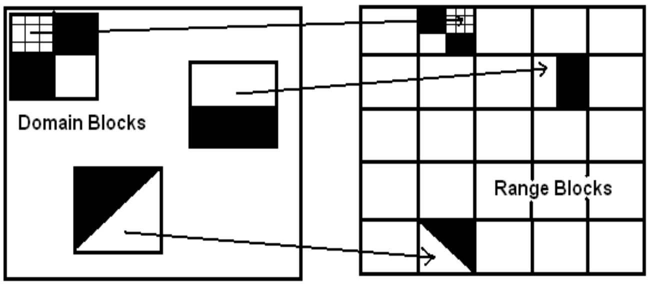

Figure 2: Mapping of Domain to Range blocks

Domain-range comparison is a three-step process.
1. One of the eight basic orientations is applied to the selected domain block.
2. The rotated domain block is contracted to match the size of the range block. The range must be smaller than the domain in order of overall mapping to be a contractive.
3. Then optimal contrast and brightness parameters are computed using least squares fitting.

As discussed in section 2, for a gray scale image *I*, if *z* denotes the pixel intensity at the position *(x, y)*, then *W* can be expressed in matrix form as follows:

$$W \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e \\ f \\ o \end{bmatrix}$$

Here coefficients *a, b, c, d, e* and *f* controls the geometrical aspects of the transformation (skewing, rotation, scaling and translation), while the coefficients *s* and *o* controls the contrast and brightness of the transformation and together make the luminance parameters. The geometrical parameters of the transformation limited to rigid translation, a contractive size-matching and one of eight orientations.

Finally representing the image as a set of transformed blocks, form a closed approximation of an original image. Minimizing the error between transformed domain block and range block will minimize the error between the original image and the approximated image.

PIFS codes of an image are having following parameters. The first parameters shows the geometrical positions of range blocks, the next column is the domain index number which uniquely locate the position of domain block using some preset parameters such as size of domain blocks, number of different domain sizes and overlapping factors. Third parameter is minimum distance between the blocks of an image, the last two parameters brightness and contrast factor *o* and *s* respectively. For recognition purpose, last four parameters of PIFS code are used.

VI. EXPERIMENTAL RESULTS

Experiments were carried out on a FERET database of known face images to verify the performance of the system. Two of the most critical requirements in support of producing reliable face-recognition systems are a large database of facial images and a testing procedure to evaluate systems. The Face Recognition Technology (FERET) program has addressed both issues through the FERET database of facial images [19].

The database consists of frontal view and other images with variations in scale, pose, expressions and illuminations of the face. Experiments were performed in two steps: generating PIFS code of database images and query image for recognition. In the first step, the PIFS code of all known faces are found and stored. The size of domain and range blocks were set to *16* X *16* and *8 X 8* pixels, respectively. In the second step PIFS codes of query image is compared with database images.

We have tested the PIFS based face recognition method on more than *30* human face images of *127* images in FERET database. In our system, one can present a query, by selecting randomly generated images from the FERET database file. Once a query face image is submitted it will retrieve face image from reference database image by using PIFS parameter & display with matching percentage. Our experimental result of a query image & three reference images shows (fig. 3) how a query image has been recognize by calculating the PIFS code & matching with a minimum distance from reference image.

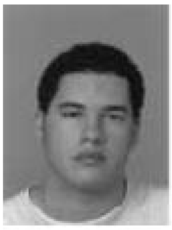

Fig 3 (a): Query Image

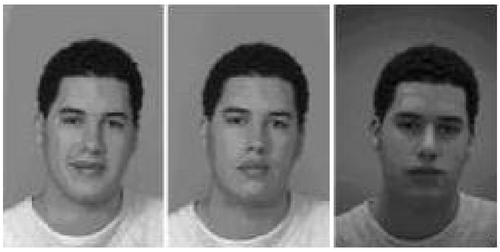

Fig 3 (b): Reference Images

The absolute pixel wise difference between the domain blocks & range blocks is called as minimum distance for that domain block with the range block. In the recognition stage, two groups of experiments were performed. In the first experiment, the known face images were presented to the system and the recognition process was carried out.

VII. ACKOWLEDGEMENT

We gratefully acknowledge the support of Sandip Foundation's, SIEM, Nashik for this research work.

CONCLUSION

In this paper we present a novel scheme for face recognition using partitioned iterated function system codes. In our method, we exploit the mapping relations between the range blocks and domain blocks in an image. Partitioned or local iterated function system is a generalization of iterated function system. This technique is based on concepts like fractals, contractivity mappings, affine transformations, *etc*. We describe a related theory with the basic concepts and construction of PIFS code. In PIFS technique the estimates of all the range blocks are obtained assuming the self similarities present in the given image. The transformed version of the domain block which is most similar to a range block is named as appropriately matched domain block for that range block. The similarity between the range block and the domain block is measured by rms metric. Thus the efficiency of PIFS technique depends on two factors The first one is the efficiency of the distortion measure and second one is the extent of similarity present in the domain blocks corresponding to the range blocks of a given image. This system is less computationally expensive than other methods like PCA. Because when a new face is added to the database, recalculation of eigenfaces are necessary in PCA based method. But in our described method we can add new faces without any such overhead.

REFERENCES

[1] Nefian AV, Hayes MH, An embedded HMM based approach for face detection and recognition. IEEE International Conference on Acoustics, Speech and Signal Processing 1999, 6:3553-3556.
[2] Wiskott L, Fellous JM, Kruger N, CVD Malsburg, Face recognition by elastic bunch graph matching. Intelligent Biometric Techniques in Fingerprint and Face Recognition 1999, Chapter 11:355-396.
[3] Teewoon Tan and Hong Yan,"Face Recognition Using the Weighted Fractal Neighbor Distance" IEEE transactions on systems, MAN and CYBERNETICS – Part C: Applications and reviews, vol. 35, no. 4, november 2005
[4] Moon H, Philips PJ, Computational and performance aspects of PCA based face recognition algorithms. Perception 2001, 30:303-321.
[5] Hang HZ, Zhang B, Huang W, Tian Q, Gabor wavelet associative memory for face recognition. IEEE Trans Neural Netw 2005, 16(1):275-278.
[6] Zhao W, chellappa R, Phillips PJ, Rosenfeld A, Face recongnition: a literature survey. ACM Comput Surv 2003, 35(4):399-458.
[7] Wohlberg BE, De Jager G, A review of the fractal image coding literature.IEEE Trans Image Process 1999, 8(12):1716-1729.
[8] Tan T, Yan H, Object recognition based on fractal neighbor distance. Signal Process 2001, 81:2105-2129.
[9] M. Barnsley, "Fractals Everywhere," Academic Press, San Diego, 1988.
[10] A. E. Jacquin, "Fractal image coding: A review," Proc. of the IEEE, vol. 81, pp. 1451-1465, 1993.
[11] Y. Fisher, "Fractal Image Compression, Theory and Application," Springer-Verlag Inc., 1995
[12] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces," Proc. IEEE Conj on Computer Vision and Pattern Recognition, pp. 586-591, 1991.
[13] P. Debevec, "A Neural Network for Facial Feature Location," UC Berkeley CS283 Project Report, December 1992.
[14] G. Neil and K. M. Curtis, "Scale and Rotationally Invariant Object Recognition Using Fractal Transformations," In Proceedings of ZCASSP-96,vol. 6, pp. 3458-3461, 1996.
[15] A. Z. Kouzani, F. He, and K. Sammut, "Fractal Face Representation and Recognition," IEEE Int'l Con& on Systems, Man and Cybernetics, vol. 2,pp. 1609-1613, 1997
[16] A. Z. Kouzani, F.He, and K. Sammut, "Face Image Matching Using Fractal Dimension," IEEE Int'l Conj on Image Processing, pp. 642-646, 1999.
[17] Suman K Mitra, C. A. Murthy and Malay K Kundu," A Study on Partitioned Iterative Function Systems for Image Compression", Fundamenta Informaticae, IOS Press.
[18] S. A. Curtis and C. E. Martin, " Functional Fractal Image Compression ",Department of Computing, Oxford Brookes University, UK.
[19] P. Jonathon Phillips and Syed A. Rizvi, " The FERET Evaluation Methodology for Face Recognition Algorithms", IEEE transactions on pattern analysis and machine intelligence, vol. 22, no. 10, October 2000.