

# A Review on Block Matching Motion Estimation and Automata Theory based Approaches for Fractal Coding

Shailesh D. Kamble<sup>1</sup>, Nileshsingh V. Thakur<sup>2</sup>, and Preeti R. Bajaj<sup>3</sup>

<sup>1</sup>Computer Science & Engineering, Yeshwantrao Chavan College of Engineering, India

<sup>2</sup>Computer Science & Engineering, Prof Ram Meghe College of Engineering & Management, India

<sup>3</sup>Electronics Engineering, G. H. Rasoni College of Engineering, India

**Abstract** --- Fractal compression is the lossy compression technique in the field of gray/color image and video compression. It gives high compression ratio, better image quality with fast decoding time but improvement in encoding time is a challenge. This review paper/article presents the analysis of most significant existing approaches in the field of fractal based gray/color images and video compression, different block matching motion estimation approaches for finding out the motion vectors in a frame based on inter-frame coding and intra-frame coding i.e. individual frame coding and automata theory based coding approaches to represent an image/sequence of images. Though different review papers exist related to fractal coding, this paper is different in many sense. One can develop the new shape pattern for motion estimation and modify the existing block matching motion estimation with automata coding to explore the fractal compression technique with specific focus on reducing the encoding time and achieving better image/video reconstruction quality. This paper is useful for the beginners in the domain of video compression.

**Keywords** --- Fractal Compression, Compression Ratio, Encoding Time, Decoding Time, Automata, Motion Estimation.

## I. INTRODUCTION

WITH the most challenging area in computer animations and multimedia technology, data compression remains a key issue regarding the cost of storage space and transmission times. Fractal Compression was first promoted by M. Barnsley which is based on Iterated Function System (IFS) [1][2]. It basically deals with the exploration of the self-similarity present in the given image. Though the Fractal coding is advantageous with respect to the compression ratio and image reconstruction quality, but it has the heavier non-acceptance related to the time elapsed for the check of similarity. It is suitable for the gray level image compression, but later some new techniques were also developed for the color image/video compression. The collage theorem [3] is the basis for the fractal transform. The collage theorem for an input image  $I$ , a new set  $W(I)$  is computed by the union of  $n$  number of sub-images, each of which is formed by applying a contractive affine transformation  $w_i$  on  $I$  as given in (1). A practical reality was given to fractal compression by Jacquin with partitioned IFS (PIFS) [3].

$$W(I) = \bigcup_{i=0}^n w_i(I) \quad (1)$$

Video compression deals with the compression mechanism for the series of image sequences. In coding, the correlation between the adjacent image frames may get explored, as well as the relativity between them may also be used in the development of the compression mechanism. Generally, the adjacent image frame does

not differ much. The probable difference lies in the displacement of the object in the given image frame with respect to the previous image frame. Grey and color videos (i.e. image sequences of gray or color image frame) are the customers for the video compression approach. Different color spaces [4] can be used for the video images from the processing point of view. With the very fast development in multimedia communication with moving video pictures, processing on color images plays a very important role. A color image is represented by 24 bits/ pixel in RGB color space format, with each color component represented by 8 bits.

Block matching motion estimation is a popular technique for many motion compensated video coding standards. Video compression standards, in general, are used for the video coding. Basic video compression standards are- Video coding standards are related to the organizations- ITU-T Rec. H.261, ITU-T Rec. H.263, ISO/IEC MPEG-1, ISO/IEC MPEG-2, ISO/IEC MPEG-4, and recent progress is H.264/AVC. In a series of the image sequence, there are spatial, temporal and statistical data redundancy that arises between frames. Motion estimation and compensation are used to reduce temporal redundancy between successive frames. Motion estimation computes the motion/movement of an object in a given image. For achieving data compression in a sequence of images, the motion compensation uses the knowledge of the motion object. Different searching techniques are available to compute the motion estimation between frames.

A finite automaton is a mathematical model used in the theoretical foundation of computer science to show acceptance or rejection of a particular string of an algorithm. Transition diagram, i.e. graph, is a convenient way of designing finite automata. Any image can be represented by a finite automaton. Because of its simple mathematical structure, a finite automation is used in fractal image/video compression. The principle of finite automata is based on the self-similarity present within the picture itself. Image compression with finite automata can also be applied to digital video sequences, which are typically represented by a series of frames or digital images [5]. The concept of finite automata is now generalized to weighted finite automata.

The remainder of this paper is organized as follows: An overview of fractal Image compression is given in section 2. Section 3 describes the related work on fractal image coding. An overview of fractal video coding is given in Section 4. Section 5 describes a related work on automata theory based coding. Section 6 focuses on work carried out on fractal video coding using block matching motion estimation. This section elaborates the different motion estimation approaches and Section 7 discusses about the quality measures associated with video processing. Section 8 summarizes the conclusion from the studied existing approaches. Finally, the paper ends with the future scope in section 9 followed by references.

## II. OVERVIEW OF FRACTAL IMAGE CODING

In general, there are two types of compression—Lossless and Lossy. In lossy data compression techniques some amount of the original data is lost during the compression process. For fast transmission of images across the internet media lossy techniques are used in World Wide Web. Examples of lossy techniques are JPEG, GIF, Wavelet, Fractal, DCT etc. In lossless data compression techniques very few amount of data is lost. Examples of lossless techniques are: TIFF, CCD RAW, etc.

Fractal coding techniques are generally applied on gray level images. For color image compression, each of red, green and blue component is compressed individually using gray image fractal coding algorithm. Fractal coding is a block-based processing technique which takes long encoding time for compression but less decoding time for decompression and it falls in the category of lossy compression technique. In basic fractal image coding, the original image is divided into small non-overlapping range block (R block) of fixed size which is nothing but a group of a collection of horizontal and vertical pixels. For each R block, find overlapping domain block (D block) of fixed size which is also a group of a collection of horizontal and vertical pixels and are generally two or four times the size of the range block. For instance, If the image of size  $I=2^N \times 2^N$  is divided into non-overlapping range block  $R_{i=0,1,2,\dots,m}$  of size  $2^n \times 2^n$ , then search for the best match overlapping domain block  $D_{j=0,1,2,\dots,n}$  of size  $2^{n+1} \times 2^{n+1}$  i.e. double the size of range block. The number of range block for a single plane is  $R_m = (2^N \times 2^N) / 2^n \times 2^n$  and the number of domain block for a single plane is  $D_n = ((2^N \times 2^N) - (2^{n+1} \times 2^{n+1}) + 1)^2$ . Domain pool  $D_p$  consists of all the transformed and under-sampled domain block such that it matches to the size of the range block. Block schematic of basic fractal coding encoder is shown in Fig. 1. The steps for basic fractal encoding algorithm [6] are as follows.

1. Divide the original image to be encoded to get the non overlapping range blocks  $R_i$ .
2. Divide the original image to be encoded to get the overlapping Domain blocks  $D_j$ .
3. Generate domain pool  $D_p$  consisting of all transformed and shrink domain blocks  $D_j^t$ .
4. For each domain block  $D_j^t$  in domain pool  $D_p$ , using least square regression method [3] given in equation (2) and (3), compute the values of contrast factor- $s$  and brightness factor- $o$  by referring  $R_i$  and  $D_j^t$  in domain pool  $D_p$ . Given a pair of range block  $R_i$  and transformed and shrink domain block  $D_j^t$  in domain pool  $D_p$  of  $n$  pixels with intensities  $r_1, r_2, \dots, r_n$  and  $d_1, d_2, \dots, d_n$  to minimize the quantity i.e R.

$$R = \sum_{i=1}^n (s \cdot d_i + o - r_i)^2$$

$$\text{where } s = \frac{[n \sum_{i=1}^n d_i r_i - \sum_{i=1}^n d_i \sum_{i=1}^n r_i]}{[n \sum_{i=1}^n d_i^2 - (\sum_{i=1}^n d_i)^2]} \quad (2)$$

$$\text{and } o = \frac{1}{n} [\sum_{i=1}^n r_i - s \sum_{i=1}^n d_i] \quad (3)$$

5. Compute error  $E(R_i, D_j^t)$  using equation (4) and Quantize factor  $s$  and  $o$  using uniform quantizer.

$$E(R_i, D_p) = \frac{1}{n} \left[ \sum_{i=1}^n r_i^2 + s(s \sum_{i=1}^n d_i^2 - 2 \sum_{i=1}^n d_i r_i + 2o \sum_{i=1}^n d_i) + o(no - 2 \sum_{i=1}^n r_i) \right] \quad (4)$$

if  $[n \sum_{i=1}^n d_i^2 - (\sum_{i=1}^n d_i)^2] = 0$  then  $s=0$  and  $o = \frac{1}{n} [\sum_{i=1}^n r_i]$

6. Search all domain blocks  $D_j^t$  in domain pool  $D_p$  for a particular range block to be encoded  $R_i$  and find the most suitable block  $D_j^t$  with minimal error  $(R_i, D_j^t) = \min E(R_i, D_p)$ .

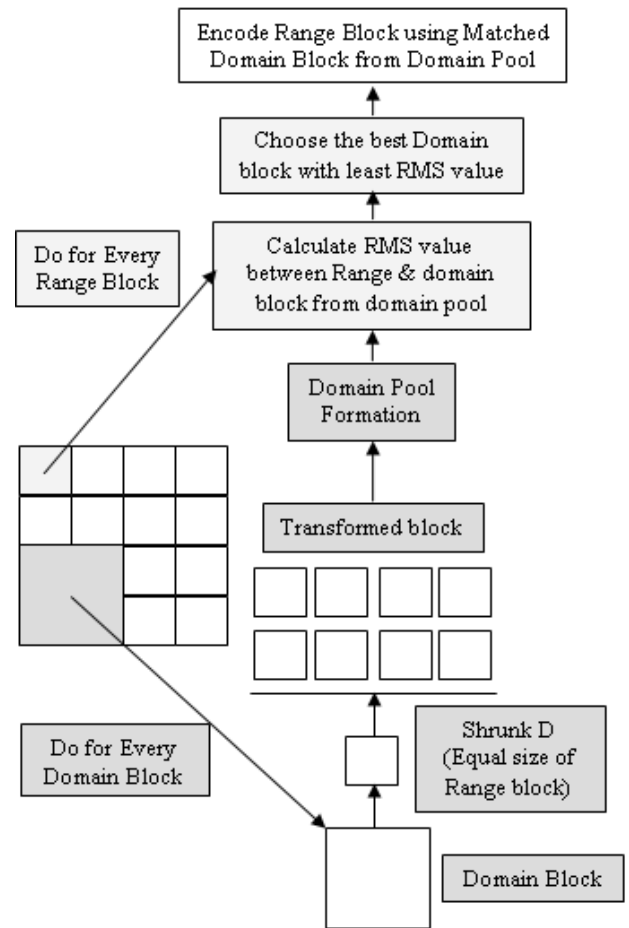


Fig. 1. Schematic of Basic Fractal Coding Encoder Mechanism.

The decoding of a compressed image can be easily achieved with a starting value  $B^{(0)}$  given in relation (5)

$$B^{(q)} = W(B^{(q-1)}) \quad (5)$$

Where  $W$  is Fractal Transformations applied on each domain block,  $q$  is number of transformations i.e. Eight affine/linear transformations and  $\{B^{(0)}, B^{(1)}, B^{(2)}, B^{(q-1)}, B^{(q)}\}$  set of sequence of transformation. The Peak-signal-noise-ratio(PSNR) value describes the image quality of the decoded image computed by using equation (6).

$$\text{PSNR} = 10 \cdot \log_{10} \left[ \frac{A^2}{\text{MSE}} \right] \quad (6)$$

Where  $A$  is amplitude of the signal and is given as  $A=2^8-1=255$  8-bit gray image and MSE is computed as follows

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_{i,j} - X_{i,j})^2 \quad (7)$$

Where  $X_{i,j}$  and  $Y_{i,j}$  are the pixel  $(i, j)$  intensities of original and decoded image respectively.

In sequentially approximating each range block by suitable domain block, the major drawback is exhaustive searching that is required for finding a best matching domain block, which leads to increase encoding time. Several approaches have been proposed for reducing encoding time. In this review paper, we present various approaches for reducing encoding time for the image as well as video fractal coding. Classification of fractal image and video coding approaches is represented using Block schematic as shown in Fig. 2.

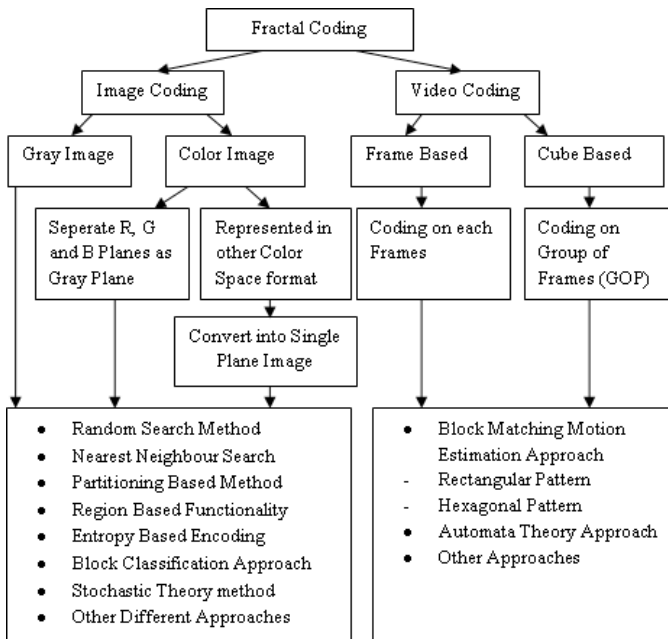


Fig. 2. Classification of Fractal Image and Video Coding Approaches.

### III. RELATED WORK ON FRACTAL IMAGE CODING

Literature available on Fractal gray image compression is very large and few report about the color image compression. Formulation of approximate nearest neighbor search in [7] is based on orthogonal projection and prequantization of the fractal transforms parameters. Ghosh et al. [8] searched domain blocks randomly for every range block to minimize the encoding time. Truong et al. [9] optimized the range blocks pool and domain blocks pool using spatial correlation to minimize the search space and searching time. The encoding algorithm in [10] based on the law of cosines. Fan and Liu [11] presented the matching algorithm based on the Standard Deviation (STD) between range blocks and domain blocks. Wang et al [12] proposed Correlation information feature to find nearest neighbor domain block for each range block.

Ghosh et al. [13] proposed an approach for fractal image coding based on an innovative concept of relative fractal coding which found to be suitable for coding multi-band satellite images. Conci and Aquino [14] used fractal dimensions for the image part classification where first the image block fractal dimension complexity is evaluated and then only parts within the same range of complexity are used for testing the better self-affine pairs. Li et al. [15] presented a kernel function clustering based on an ant colony algorithm. It automatically realizes classification of the domain block. Hartenstein et al. [16] presented bottom-up region merging approach where regions are merged on the basis of collage error. Belloulata and Konrad [17] proposed an approach to fractal image coding that permits region-based functionalities where images are coded region by region according to a previously-computed segmentation map. The method in [18] is particularly well suited for use with highly irregular image partitions for which most traditional (lossy) acceleration schemes lose a large part of their efficiency. Franco and Mala [19] presented an algorithm for adaptive image partitioning achieving designated rates under a computational complexity constraint. Wang [20] proposed a graph based image segmentation algorithm used to divide the image into the different logical area and each logical area is coded into adaptive threshold quadtree partitioning approach.

Hassaballah et al. [21] minimized domain pool size on the basis of entropy value of each domain block. Domain pool reduction is

parameterized and non-adaptive by allowing an adjustable number of domains to be excluded from the domain pool based on the entropy value of the domain block. He et al. [22] utilized one-norm of a normalized block to minimize the domain pool search space, in which the search process might be terminated early, and thus remaining domain blocks could be safely discarded. Rowshanbin et al. [23] used a special characteristic vector to classify the domain blocks to minimize the searching time. The approach in [24] uses the minimum distortion and variance difference between the range block and domain block to minimize the domain pool and searching time. To speed up the encoding, in [25], the classification features are used to classify the image blocks. Xing et al. [26] used hierarchical partitioning to classify the domain pool. Fuzzy pattern classifier is utilized in [27], to classify the original image blocks. Qin et al. [28] sorted domain pool on a number of hopping and the variance of continuing positive and negative pixels in the given block. The approach in [29] reduces the memory requirement, and speeds up the reconstruction. In Stochastic image compression using Fractals [30], stochastic image coding based on the fractal theory of iterated contractive transformations is discussed. Fractal image compression based on the theory of iterated function system (IFS) with probabilities is explained in [31]. Gungor and Ozturk [32] discussed a hash function based image classification technique. No search fractal image compression in DCT domain is discussed in [33]. Eugene and Ong [34] discussed two pass encoding scheme to speed up the encoding process. The approach in [35] is based on computing the gray level difference of domain and range blocks. Peng et al. [36] obtained the range blocks by partitioning the image using adaptive quad trees. A direct allocating method to predict the desired transformation for similarity measure is discussed in [37]. Liu et al. [38] presented the entropy based encoding algorithm. Melnikov [39] presented the method of acceleration of the image fractal coding. Distasi et al. [40] proposed an approximation error based approach to classifying the blocks. The fundamental idea of this approach consists in deferring range and domain comparisons, based on feature vectors.

A preset block  $\tilde{d}$ , as a temporary replacement with which range and domain blocks are compared.

There are many color spaces [4] available to represent the color images. To display the image on the monitor, the RGB color space is generally used. For color image compression, the RGB model is best suited because it provides the highest correlation [41]. For color image compression on square architecture, the fractal coding is applied on different planes of a color image independently by treating each plane as a gray level image. This approach is Straight Fractal Coding or Separated Fractal Coding (SFC) [42]. Work related to fractal coding of color images reported in the literature are: Hurtgen et al. discussed a fractal transform coding of color images in [43]. To exploit the spectral redundancy in RGB components, the root mean square error (RMSE) measure in grey-scale space is extended to 3-dimensional color space for fractal-based color image coding in [44], where it is claimed that 1.5 compression ratio improvement can be obtained using vector distortion measure in fractal coding with fixed image partition as compared to separate fractal coding in RGB images. Comparative study of fractal color image compression in the  $L^*a^*b^*$  color space with that of Jacquin's iterated transform technique for 3-dimensional color is presented in [45], where it is claimed that the use of uniform color space has yielded compressed images with less noticeable color distortion than other methods. Li et al. [46] presented fractal color image compression scheme based on the correlation between the three planes of RGB color space. Giusto et al. presented an approach for color image coding based on the joint use of the  $L^*a^*b^*$  color space and Earth Mover's Distance in [47]. Thakur et al. [6] discussed a Fractal compression technique, which is basically a searching technique based on self-similarity search within an image and elaborated basic steps

required for Fractal coding technique i.e. partitioning into rang/domain blocks, searching each range block with all domain blocks and stores the values of best transformations. Thakur and Kakde [51] proposed a modified fractal coding approach on spiral architecture to optimize domain blocks using local search. One of the major challenges in fractal coding technique is to optimize/reduce the size of domain pool. Reducing the number of domain blocks in a domain pool reduces encoding time in fractal coding technique.

IV. OVERVIEW OF FRACTAL VIDEO CODING

There are two methods for fractal video: coding-Frame based and Cube based fractal video coding. In frame based coding, each single frame of image is partitioned into a range and domain square block, which is already described in section 2, and then each frame is coded based on inter-frame and intra-frame coding to reduce the redundant part of data within a frame and between two frames using fractal transformations. The inter-frame and intra-frame coding are discussed in section 4.1. Every range block of the current frame is encoded by the domain block from the previous frame i.e. encode every range block by using inter-frame similarity as shown in Fig. 3. The error occurs in the latter/next frames due to the inter-frame similarity, i.e use of domain block from the previous frame, and delay occurs between two successive frames during the processes of decoding, which are the major drawbacks in frame based video coding. This method is advantageous in achieving a high compression ratio and thus it is used in video transmission through the internet/www media.

In cube based coding, The video sequence to be coded is first combined into a group of frames(GoF) and then each GoF is partitioned into a set of range and domain cubes as shown in Fig. 4. For instance, image can be viewed as 3D /cubic digital image of size  $I=2^N \times 2^N \times 2^N$ , which is divided into non-overlapping range block  $R_{i=0,1,2,...,m}$  of size  $2^n \times 2^n \times 2^n$  then search for the best match overlapping domain block  $D_{j=0,1,2,...,n}$  of size  $2^{n+1} \times 2^{n+1} \times 2^{n+1}$  i.e double the size of range block . The number of range blocks for a single plane is  $R_m = (2^N \times 2^N \times 2^N) / 2^n \times 2^n \times 2^n$  and the number of domain blocks for a single plane is  $D_n = ((2^N \times 2^N \times 2^N) - (2^{n+1} \times 2^{n+1} \times 2^{n+1}) + 1)^2$ . Domain pool  $D_p$  consists of all the transformed and under-sampled domain blocks such that it matches the size of the range blocks as shown in Fig. 5. If  $N = 2$  and  $n = 1$ , then for a cubic digital image of size  $I=4 \times 4 \times 4$  having a number of pixels 64, the number of range blocks  $R_m = 8$ , having a number of pixels in each range block that is 8, and the number of domain blocks  $D_n = 1$ , having a number of pixels in domain block that is 64. This method is advantageous in obtaining a high quality of decoded image in receiving end but disadvantageous in achieving a low compression ratio.

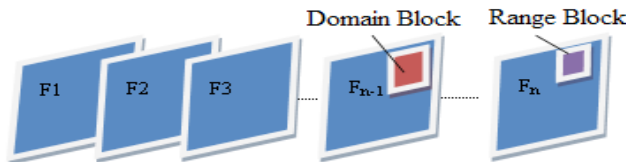


Fig. 3. Range and Domain Blocks Formation in Frame based Video Coding.

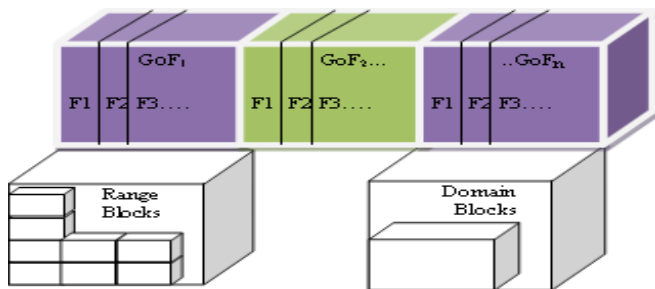


Fig. 4. Range and Domain Cubes Formation in Cube based Video Coding.

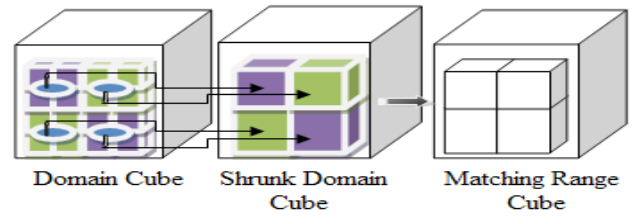


Fig. 5. Domain Cube Matching with Range Cube.

The basic fractal encoding algorithm for still images given in section 2 is easily extended to fractal video coding. The steps for obtaining frame based fractal video coding are as follows.

1. Extract image sequences from video  $F_f = \{F_1, F_2, \dots, F_n\}$ .
2. For each frame  $F_f \{ 2 \leq f \leq n \}$  do
3. Apply fractal coding for still image  $F_{f-1}$  to compute decoded image  $DF_{f-1}$ .
4. Divide the frame  $F_f$  to get the non overlapping Range block  $R_i$ .
5. Divide the decoded frame  $DF_{f-1}$  to get the overlapping domain block  $D_j$ .
6. Generate domain pool  $D_p$  from decoded frame  $DF_{f-1}$  consisting of all transformed and shrink domain blocks  $D_j^t$ .
7. For each domain block  $D_j^t$  in domain pool  $D_p$ , using least square regression method [3] given in equation (2) and (3) compute the values of contrast factor-s and brightness factor-o by referring  $R_i$  and  $D_j^t$  in domain pool  $D_p$ . Given a pair of range block  $R_i$  and transformed and shrink domain block  $D_j^t$  in domain pool  $D_p$  of n pixels with intensities  $r_1, r_2, \dots, r_n$  and  $d_1, d_2, \dots, d_n$  to minimized the quantity i.e R.

$$R = \sum_{i=1}^n (s \cdot d_i + o - r_i)^2$$

8. Compute error  $E(R_i, D_j^t)$  using equation (4) and Quantize factor s and o using uniform quantizer.
9. Searching all domain block  $D_j^t$  in domain pool  $D_p$  for a particular range block to be encoded  $R_i$  and find the most suitable block  $D_j^t$  with minimal error  $(R_i, D_j^t) = \min E(R_i, D_p)$ .

Similarly a basic fractal encoding algorithm for still image is easily extended for obtaining cube based fractal video coding are as follows.

1. Extract image sequences from video  $F_f = \{F_1, F_2, \dots, F_n\}$ .
2. Frame sequence is divided into  $GoF_i = \{GoF_1, GoF_2, \dots, GoF_n\}$ .
3. For each  $GoF_i$  do
4. Divide the  $GoF_i$  to get the non overlapping range cube block  $R_i$ .
5. Divide the  $GoF_i$  to get the overlapping domain cube block  $D_j$ .
6. Generate domain pool  $D_p$  from  $GoF_i$  consisting of all transformed and shrink domain blocks  $D_j^t$ .
7. For each domain block  $D_j^t$  in domain pool  $D_p$ , using least square regression method [3] given in equation (2) and (3) compute the values of contrast factor-s and brightness factor-o by referring  $R_i$  and  $D_j^t$  in domain pool  $D_p$ . Given a pair of range block  $R_i$  and transformed and shrink domain block  $D_j^t$  in domain pool  $D_p$  of n pixels with intensities  $r_1, r_2, \dots, r_n$  and  $d_1, d_2, \dots, d_n$  to minimized the quantity i.e R .

$$R = \sum_{i=1}^n (s \cdot d_i + o - r_i)^2$$

8. Compute error  $E(R_i, D_j^t)$  using equation (4) and Quantize factor s and o using uniform quantizer.

9. Searching all domain blocks  $D_j^i$  in domain pool  $D_p$  for a particular range block to be encoded  $R_i$  and find the most suitable block  $D_j^i$  with minimal error  $(R_i, D_j^i) = \min E(R_i, D_p)$ .

### A. Intra-frame and Inter-frame Coding

In video coding there are two ways to reduce the data present in the frames. Firstly, spatial redundancy elimination, which is called intra-frame coding, in which the frames are coded individually as done in JPEG compression technique. Within individual frame coding, similar data part can be coded with fewer bits per pixels than the original data part, therefore reducing bits per pixel is a minor loss in noticeable visual quality of individual frame coding. On the other hand, temporal redundancy elimination, which is called inter-frame coding, in which the redundant data are eliminated between the frames as done in MPEG compression technique. Inter-frame coding finds the difference between the previous frame and current frame and stores only difference i.e. displacement of pixels instead of complete frames. The displacements of pixels are estimated by using a well-known technique called block matching motion estimation. Block matching motion estimation techniques are used to find out the motion vectors in a frame and then the displacement of pixels block identified by motion estimation technique is coded. Later this block is considered as a previous frame and the next frame in a sequence is considered as a current frame for finding out the difference. This coding process is repeated for all the remaining frames in a video sequence. Fig. 6 shows the block schematic of intra-frame and inter-frame coding techniques.

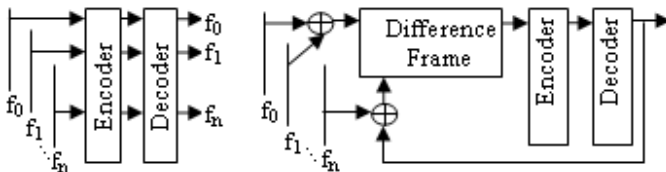


Fig. 6. Intra-frame (Spatial) and Inter-frame (Temporal) Coding Technique.

### B. Video Coding Standards

Video compression standards, in general, are used for the video coding. Basic video compression standards are- Video coding standards are related to the organizations are- ITU-T Rec. - H.261, H.263, ISO/IEC- MPEG-1, MPEG-2, MPEG-4, and H.264/AVC. Summary of video coding standards with their properties are given in Table I. Block matching motion estimation algorithms play an important role in designing video coding standards. Video coding standards consist of motion estimation algorithm, encoding mechanism and decoding mechanism to eliminate redundant data.

### C. Preliminaries of Automata Theory

In general, a finite automaton is a simple mathematical model used to recognize the strings generated by regular expression notation. In image / video compression technique, finite automata are used to represent the entire image and the address of each subimages of the entire image is specified by the regular expression i.e.  $\Sigma^* = \{0, 1, 2, 3\}^*$  where 0, 1, 2 and 3 are the address of the subimages. Finite automata coding process is similar to the fractal coding process by extending finite automata model with the weighted finite automata (WFA) model. WFA is an extended version of finite automata in which a weighted transition is associated between the two states.

A WFA  $M = (Q, \Sigma, W, I, F, q_0)$  consists of a finite set of states i.e.  $Q = \{q_0, q_1, q_2, \dots, q_n\}$ , a finite set of input symbol/quadrant address i.e.  $\Sigma = \{0, 1, 2, 3\}$  for Quadtree WFA,  $\Sigma = \{0, 1\}$  for Bintree WFA and  $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$  for Nonatree WFA, the weight function between two states i.e.  $W \in Q \times Q \rightarrow R$ , for  $W(q_0, q_1) = R$  where  $q_0, q_1 \in Q$  and  $R$  is the real number i.e. weight between state  $q_0$  and  $q_1$ , the initial configuration (I) of states  $Q \rightarrow R$  and indicates which states corresponds to the entire image  $I(q_0)=1$  and  $I(q_i)=0, q_0 \neq q_i$  where  $q_0, q_i \in Q$  and  $i=1, 2, 3, \dots, n$ , the final configuration (F) of states  $Q \rightarrow R$  e.g.  $F(q_0) = f(\epsilon)$  where  $q_0 \in Q$  and  $f(\epsilon)$  is the average intensity (Greyness) of the entire image, and the initial state  $q_0$  of the WFA means the entire original image i.e.  $q_0 \in Q$ .

The extended version of weighted finite automata is called as extended weighted finite automata (EWFA), which is similar to WFA in which all the subimages in EWFA are transformed using transformation function i.e. scaling and rotation etc. Similarly to WFA, EWFA is also used to store and compress data represented as an image/matrix. In EWFA the numbers of states to store the subimages/matrix data are less as compared to WFA and therefore memory space required to store the states are less.

TABLE I  
SUMMARY OF VIDEO CODING STANDARDS WITH OBJECTIVES/ PROPERTIES

Standard	Organization	Standards Properties
MPEG-1	ISO/IEC	Main application is multimedia video storage, video frame rate - 24 to 30 fps, bit rate- 1.5 Mbps, image format- CIF and no interlace.
MPEG-2	ISO/IEC	TV resolution- NTSC:704 x 480 Pixel, digital cable TV distribution similar to NTSC, PAL, SECAM at 4 -8 Mbps, HD television at 20 Mbps and interlace
MPEG-4	ISO/IEC	Object based coding and used in wide range of applications i.e. TV film, video communication, video cameras, Internet streaming, mobile phones etc. with choices of interactivity, scalability, error resilience, etc.
H.261 and H.263	ITU-T	Video conferencing applications over ISDN telephone lines
H.264	ITU-T and ISO/IEC	<ul style="list-style-type: none"> <li>Broadcast over cable, modems, ADSL, terrestrial etc.</li> <li>Optical / magnetic storage devices and DVDs storages.</li> <li>Video-on-demand or streaming service and Multimedia messaging service (MMS) over ISDN, LAN, Ethernet, modems, wireless and mobile networks etc, or their combinations.</li> </ul>

### V. RELATED WORK ON AUTOMATA THEORY

Weighted Finite Automata (WFA) is a generalization of finite automata by attaching real numbers as weights to states and transitions proposed by Culik and Kari [49-50]. WFA provides a powerful tool for image generation and compression. The inference algorithm for WFA subdivides an image into a set of non-overlapping range images and then separately approximates each one with a linear combination of the domain image. A new predictive video-coding technique [50], using fractal image compression for intra-frame coding and second order geometric transformations for motion compensation in inter-frame predictive coding, is proposed. For motion compensation second order geometric transformations, compensating for translation, rotation, zooming, uneven stretching, and any combination of these, has been used. The decoded images can be displayed at arbitrary resolution without blockiness, which, together with the very high compression ratio achievable, is the most important advantage of the fractal-based image coding technique over the standard discrete cosine transform (DCT)-based image coding [50]. WFA is one of the techniques that have been used to compress digital images [51-52]. WFA represents an image in term of a weighted finite automaton with a very good

compression ratio. WFA is based on the idea of fractal that an image has self-similarity in itself. In this case, the self-similarity is sought from the symmetry of an image, so the encoding algorithm divides an image into multi-levels of quad-tree segmentations and creates an automaton from the sub-images [53]. As the developing of the fractal image compression, the fractal coding method has been applied in video sequence compression [54], for instance, the famous hybrid circular prediction mapping and non-contractive inter-frame mapping [55]. The circular prediction mapping / non-contractive inter-frame mapping combines the fractal coding algorithm with the well-known motion estimation and compensation algorithm that exploits the high temporal correlations between adjacent frames. WFA codec is modified such that it compresses the video at low bit-rates. The video is the sequence of frames (images). Here a hierarchical motion compensation (MC) [56] and bin-tree based WFA codec are integrated to exploit the correlation between successive frames [57]. The advantage is that WFA encoding can replace other transformation coding in video applications. At low bit-rates, WFA encoded images are typically half the size of comparable JPEG images. Video coding scheme based on bit-plane modeling and GFA representation is used to explore the inter-frame, inter-bit plane and inter-level similarities present in the video. To form a generalized finite automata-based compact representation of video sequence, the GFA modeling takes advantages of the binary fractal similarity of the video sequence in the wavelet domain. After exploring the similarities we get compact generalized finite automata representation of video [58-59]. Such schemes significantly outperform the H.26X [60] series coding schemes in rate distortion performance and retain an acceptable perceptual quality of the video.

A generalized finite automaton [61] is used to encode/decode the images automatically. When the GFA is combined with the wavelet transform technique, the GFA is constructed in such a way that each state represents one wavelet function. So this method combines the advantages of both, the classical wavelet compression method and GFA. While encoding the image, GFA doesn't have to solve equations. So encoding may take significantly less time. Decoding of images can also be done more quickly. This method of GFA allows any combination of rotations, flips, and complementation of the quadrant image [62]. Quad-tree based EWFA coding is used in fractal color video compression technique [66]. The quadtree partitioning scheme is used to specify the address of each sub-images i.e. a complete quadtree represents a pyramidal structure of an entire image that is required for the EWFA encoding and decoding process. The concept of intra-frame coding i.e. individual frame coding is used to encode the number of frames in a video. In Quadtree based EWFA coding, each color frame is converted into the  $YCbCr$  color space and then converted into gray scale [67, 68]. Each gray scale frame is divided into a fixed sized block ( $2^n \times 2^n$ ) based on quadtree partitioning to represent the address of each pixel.

## VI. RELATED WORK ON MOTION ESTIMATION

In circular prediction mapping and non-contractive inter-frame mapping, each range block is motion compensated by a domain block in the previous frame, which is of the same size as the range block even though the domain block is always larger than the range block in conventional fractal image codec. The main difference between circular prediction mapping and non-contractive inter-frame mapping is that circular prediction mapping should be contractive for the iterative decoding process to converge, while non-contractive inter-frame mapping needs not be contractive since the decoding depends on the already decoded frames and is non-iterative [66]. Recently, Wang [67] proposed a hybrid fractal video compression algorithm, which merges the advantages of a cube-based fractal compression method and a frame-based fractal compression method; in addition, an adaptive partition instead of fixed-size partition is discussed. The

adaptive partition [68] and the hybrid compression algorithm exhibit, relatively, the high compression ratio for image [68] and the video conference sequences [67]. In conclusion, a fractal image codec performs better in terms of very fast decoding process as well as the promise of potentially good compression [69-73]. But at present, the fractal codec is not standardized because of its huge calculation amount and slow coding speed. In order to alleviate the above difficulties, a block-matching motion estimation technology [74, 75] can be used to improve the encoding speed and the compression quality [76].

Block-matching motion estimation is a vital process for many motion-compensated and video coding standards. Motion estimation could be very computationally intensive and can consume up to 60%-80% of the computational power of the encoding process [77]. So research on efficient and fast motion estimation algorithms is significant. Block matching algorithms are used widely because they are simple and easy to be applied. In the last two decades, many block matching algorithms are proposed for alleviating the heavy computations consumed by the brute-force full search algorithm which has the best prediction accuracy, such as the new three-step search [78], the four-step search [79], the block-based gradient descent search [80], the diamond search [81], the cross-diamond search [82], etc. All these searches employ rectangular search patterns of different sizes to fit the center-biased motion vector distribution characteristics [83-84]. Hexagon-based search employs a hexagon-shaped pattern and results in fewer search points with similar distortion [85]. Block-matching algorithm called Novel Cross-Hexagon Search algorithm is proposed in [75]. It uses small cross-shaped search patterns in the first two steps before the hexagon-based search and the proposed halfway stop technique [74]. It results in higher motion estimation speed on searching stationary and quasi-stationary blocks. The traditional algorithms use all the pixels of the block to calculate the distortions that result in heavy computations. Modified Partial Distortion Criterion [86] that uses certain pixels of the block, which alleviates the computations and has similar distortion can be used. New Cross-hexagon search algorithm (NHXS) proposed in [87-88] consist of two cross search patterns and hexagon search patterns which are similar in [75] for fast block matching Motion Estimation. This search technique is a frame based fractal video compression technique that helps to reduce the encoding time and increases the compression quality in fractal coding.

### A. Motion Estimation Algorithms

Motion estimation used in the area of video application such that video segmentation, object/video tracking, and video compression. Motion estimation means the displacement of pixels position from one frame to another frame which gives the best motion vector. For estimating a motion in a video sequence, block matching algorithms (BMA) are widely used in most of the video coding standards. In BMA a frame is divided into a non overlapping block and for that block motion vector is estimated. A motion vector is computed by finding best suitable matched block between previous frame-f and the next frame-f+1 as shown in Fig. 7. Motion vector for the block  $B_f(x, y)$  is computed as  $(+1, -1)$  i.e.  $MV-B_f(x, y)=(+1, -1)$ .

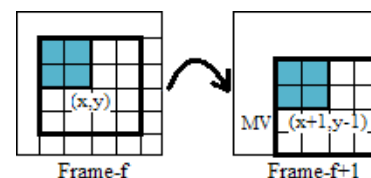


Fig. 7. Block Matching Motion Vector Estimation.

Still, research is going on for efficient and fast block matching motion estimation. Different types of block matching motion estimation algorithms are described below.

### 1) Full Search Motion Estimation

This algorithm also called as exhaustive search algorithm because motion vector is computed after a complete window of size  $(2w+1)*(2w+1)$  is exhaustively searched for the purpose of best matching of block size  $n \times n$  pixels as shown in Fig. 8(a). Due to the exhaustive search, this algorithm gives a better accuracy in searching the best matching block. The major drawback of this algorithm is to require a number of computations for the best matching block. The computational complexity of this algorithm is very high when the window size is too large and block size is too small.

### 2) Three Step Search

Koga et al. [89] proposed a robust and very simple three step search motion estimation algorithm. This algorithm is one of the most popular algorithms for the low bit rate application because of its efficient performance. Computational cost of this algorithm is low as compared to the full search algorithm. First step is to define a search window size for searching the best match. In the first step, plot nine points in the search window at the equal distance of step size. In the second step, the step size is divided by 2 if minimum block distortion measure (BDM) point is one of the nine point of search window and consider this point as a center point in the third step. This process is repeated again until the step size is smaller than one. This complete process is shown in Fig. 8(b). If the step size is  $s=7$  then the number of checking points required for TSS is 25 using equation given below. Number of checking points =  $1 + 8 \log_2(s + 1)$  where  $s$  is step size of the search window.

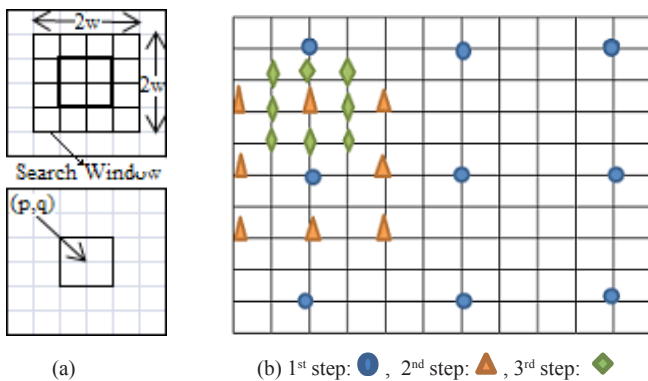


Fig. 8. (a) Full Search (b) Three Step Search.

### 3) 2D Logarithmic Search

This algorithm was originally proposed by Jain and Jain [90]. In this search strategy, an initial search window size is defined in the central area of the image. As compared to full search motion estimation algorithm, instead of searching all pixels in a complete search window, the search is done in five different directions which contain north, south, east, west and central direction as shown in Fig. 9(a), where Initial window: ●, Step size divided by 2: ▲, Final step: ◆. If the minimum BDM point is found at any of these five different directions then this direction is considered as a center of the search window for the next step in which search window area is divided by 2 and this process is repeated until the search window area is converted into  $3 \times 3$  window size. At last, all the nine points are searched in the  $3 \times 3$  search window and found minimum BDM point corresponds to best matching position that gives the motion vector i.e. block coordinates.

### 4) Orthogonal Search

A. Puri et al. [91] introduced an orthogonal search algorithm (OSA) based on the combination of three step search (TSS) and 2D logarithmic search algorithm. This algorithm performs firstly a

horizontal search with 3 checking points and secondly a vertical search with 3 checking points including minimum BDM point as a center of the previous horizontal search. The step size is divided by 2 and this process is repeated until step size is one as shown in Fig. 9(b), where 1<sup>st</sup> step: Horizontal search ●, Vertical search ▲, 2<sup>nd</sup> step: Horizontal search ■, Vertical search ◆, 3<sup>rd</sup> step: Horizontal search □, Vertical search ○. If the step size is  $s=7$  then the number of checking points required for OSA is  $3+2+2+2+2=13$  using equation given below. Number of checking points =  $1 + 4 \log_2(s + 1)$ , where  $s$  is step size of the search window.

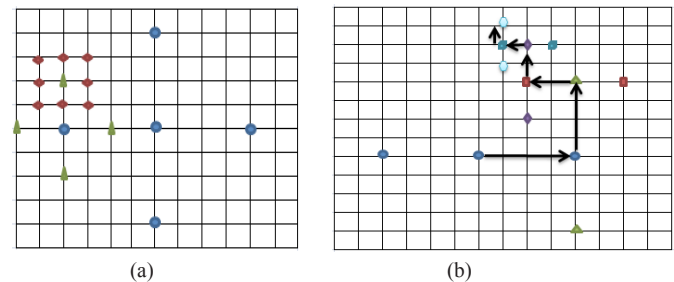


Fig. 9. (a) 2D Logarithmic Search (b) Orthogonal Search.

### 5) Cross search

M. Ganbari [92] proposed a cross search pattern algorithm (CSA) consisting of 5 checking points placed in a cross shape pattern (X). In each step find minimum BDM point then step size is divided by 2, consider this point as a center and place 4 points in cross shaped pattern across the center point. In the final step, as the step size is reduced to one, place cross search pattern (+) if minimum BDM point of the previous step found at any one of the checking point i.e. center, upper left corner and lower right corner otherwise place cross search pattern (X) as shown in Fig. 10(a), where 1<sup>st</sup> step: ●, 2<sup>nd</sup> step: ■, 3<sup>rd</sup> step: ▲, 4<sup>th</sup> step: ○. If the step size is  $s=7$  then the number of checking points required for cross search algorithm (CSA) is  $5+4+4+4=17$  using equation given below. Number of checking points =  $5 + 4 \log_2(s)$ , where  $s$  is step size of the search window.

### 6) Binary Search

Binary search (BS) Algorithm is used in MPEG-Tool [93] for block matching motion estimation shown in Fig. 10(b), where 1<sup>st</sup> step: ●, 2<sup>nd</sup> step: ◆ best case, 2<sup>nd</sup> step: ▲, average case, 2<sup>nd</sup> step: ■, Worst case. Firstly this algorithm divides the 9 checking outer points of the search window into small 9 search windows and, if minimum BDM point is found at any one of the search windows, then performs a search operation on corresponding search window. If minimum BDM point is found at the center, corner and middle of the search window then the number of checking points required for BSA is  $25+9=33$  for the worst case,  $8+9=17$  for the best case and  $14+9=23$  for the average case respectively. The pixels on the blue lines or the pixels between the search windows are not considered for searching.

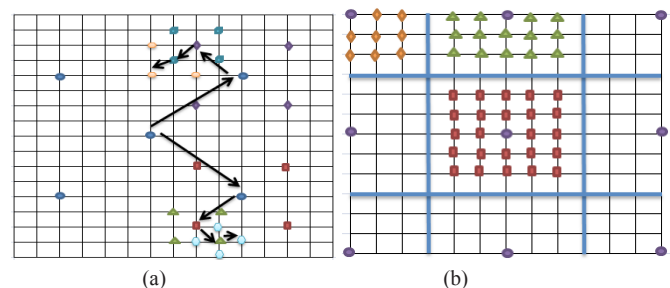


Fig. 10. (a) Cross Search (b) Binary Search.

7) *New Three Step Search*

R. Li et al. [78] proposed a new three-step search (NTSS) algorithm consisting of two searching patterns, i.e. center biased checking pattern and halfway-stop technique (step size divided by 2) like three step search algorithm as shown in Fig. 11. This algorithm initially defines a search window size for searching the best match and plots 17 points on that search window size as shown in Fig. 11. From all the 17 points, plot 9 points on the 3x3 grid in the inner/central area of search window & rest of the eight points plot on the 9x9 grid in the outer area of the search window. If the minimum block distortion measure is found at the centre point of the search window then stop/halt searching, otherwise goto step 2. In step 2, if one of the central neighboring points on 3x3 grid is found to be minimum BDM point then go to step 3, otherwise goto step 4. In step 3, if minimum BDM point is found at the corner point or middle point on the 3x3 grid in the inner/central area of the search window of the horizontal and vertical axis, then consider these points are center point and plot/search additional 5 or 3 points respectively in the search window. In step 4, if minimum BDM point is found at corner points and middle points of every two corners on the 9x9 grid in the outer area of search window, then step size is divided by 2, this process is repeated until the step size is smaller than one. In this algorithm, the number of check points required for NTSS is 17 for the best case, 20 or 22 for average case and 33 for the worst case.

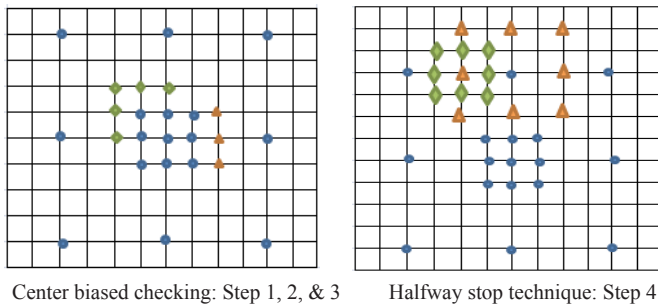


Fig. 11. New Three Step Search Motion Estimation

8) *Four Step Search*

Lai-Man Po et al. [79] proposed a novel four-step search (FSS) algorithm for block matching motion estimation. The performance measure of this search as compared to other search algorithm is better than TSS and similar to NTSS. Instead of the 9x9 search window in NTSS, this algorithm uses a central biased search pattern with 9 checking points on a 5x5 search window. If minimum BDM point is found at the center of 5x5 search window, then search window size is reduced to 3x3 window size i.e final step of FSS, otherwise goto next step. In next step, if minimum BDM point is found at any one of the four corners of the horizontal and vertical axis or midpoints between the two corners of search window, then check additional 5 points or 3 points in the search window. In the final step, a search window of size 5x5 is reduced to 3x3 window size and minimum BDM point is considered as a motion vector as shown in Fig. 12(a), where 1<sup>st</sup> Step: ●, 2<sup>nd</sup> Step: ■, 3<sup>rd</sup> Step: ▲, 4<sup>th</sup> Step: ▼. If the step size of the final step is greater than one, then again another FSS is performed with the final step of the previous FSS is the first step of the another FSS. The number of checking points required for FSS algorithm is  $9+8 = 17$  for the best case,  $9+5+5+8 = 27$  for the worst case, and  $9+3+3+8=23$  for average case if minimum BDM point found at center point in the first step, corner point in the second and third step and midpoint in the second and third step of the search window respectively.

9) *Block- Based Gradient Descent Search*

L.K.Liu et al. [80] proposed a block-based gradient descent search (BBGDS) algorithm based on central biased search pattern of 9 checking points. This algorithm performs an unrestricted search within a search window with a step size of one in each step of block generated search pattern. If minimum BDM point is found at the corners of the horizontal and vertical axis or mid points of two corners, then checked additional 5 or 3 points respectively in the search window. If the minimum BDM point is found at the center of current block searched pattern, then searching will stop or, if the search pattern reached to the search window boundary then also stop searching. This algorithm performs better for small motions. Search pattern of block based gradient descent search algorithm are shown in Fig. 12(b), where Upward MV: Step 1: ●, Step 2: ▲, Step 3: ◆.

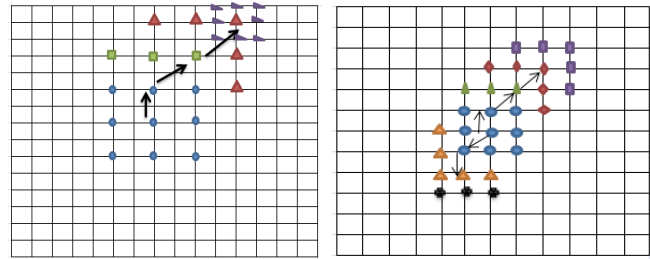


Fig. 12. (a) Four Step Search (b) Block-Based Gradient Descent Search.

10) *Diamond Search*

S. Zhu et al. [81] proposed a two diamond shape search (DS) pattern called large diamond shape search pattern (LDSP) and small diamond shape search pattern (SDSP) consisting of nine and five checking points including center point to form a diamond shape respectively. This search initially starts with 9 checking points i.e. LDSP, continue searching with this LDSP to form a new LDSP until the minimum BDM point found on the center of the LDSP and then shifted a large diamond shape search pattern to small diamond shape search pattern. In the final step of this search, the minimum BDM point found in SDSP is considered as a final motion vector as shown in Fig. 13(a), where LDSP: 1<sup>st</sup> Step: ●, 2<sup>nd</sup> Step: ◆, 3<sup>rd</sup> Step: ▲, 4<sup>th</sup> Step: ■, SDSP : 5<sup>th</sup> Step: ● Final MV. The number of checking points required for Diamond search algorithm is 13 for the best case. The performance based on the checking points computation of diamond search algorithm is closely related to efficient three-step search algorithm (ETSS) and better than three step search (TSS), four-step search (FSS) and block-based gradient descent search algorithm.

11) *Cross Diamond Search*

C.H.Cheung et al. [82] proposed a cross diamond shape search pattern algorithm based on halfway stop technique for block matching motion estimation. This search algorithm initially starts with 9 checking points cross shape (+) search pattern instead of diamond shape search pattern in the diamond search algorithm. The performance based on checking points of this algorithm is better than the diamond search algorithm. In the first step, if minimum BDM point is found at the center of large cross shape search pattern (LCSP) i.e. 9 checking points, then stop searching. In the second step, search operation is performed on small cross shape search pattern (SCSP) i.e. 5 checking points. If minimum BDM point is found at the center of the SCSP, then stop searching, otherwise in step three check another extra two points nearer to the minimum BDM point found at the corner points of SCSP. If minimum BDM point found at the center of newly created SCSP, then stop searching, otherwise perform a diamond search in the final step as shown in Fig. 13(b), where LCSP: 1<sup>st</sup> Step: ●, SCSP: 2<sup>nd</sup>



Step: ●, 3<sup>rd</sup> Step: ◆, Diamond search LDSP : ▲ & ■, SDSP : Final Step: ○ Final MV.

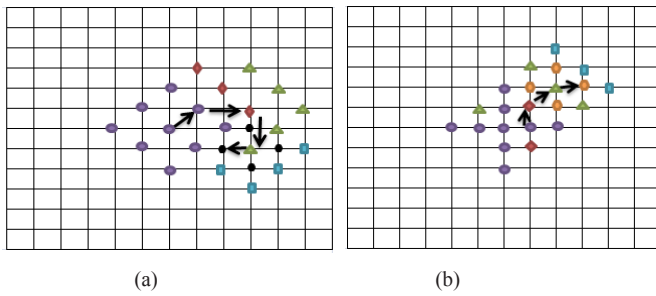


Fig. 13. (a) Diamond Search (b) Cross Diamond Search.

### 12) Hexagon Search

Ce Zhu et al. [85] introduced an hexagonal search algorithm for motion estimation, which consists of two hexagon search patterns, Large Hexagon Search Pattern (LHSP) with 7 checking points and Small Hexagon Search Pattern (SHSP) with 5 checking points including center point of the hexagon as shown in Fig. 14(a). This search technique initially starts with LHSP, if minimum BDM point is found at the center of LHSP, then LHSP is shifted to SHSP and new minimum BDM point found in SHSP is the final motion vector of the search, otherwise minimum BDM point found acts as a center of LHSP and check three more points to form a new LHSP, this step is repeated until minimum BDM point found at the center of the new generated large shape pattern hexagon and then lastly shifted LHSP into SHSP. The number of checking points required for the hexagonal search is 11 for the best case. Search pattern of hexagon search algorithm is shown in Fig. 14(b).

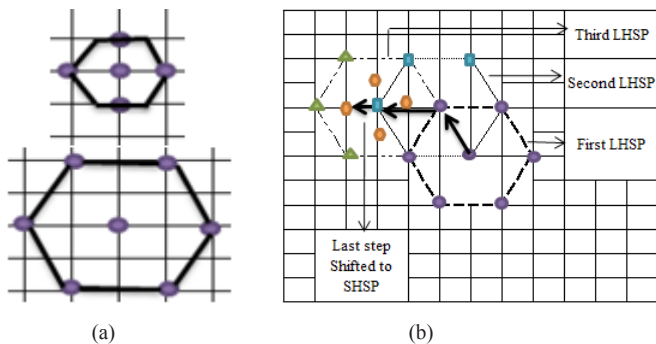


Fig. 14. (a) Large Hexagon Search Pattern (LHSP) and Small Hexagon Search Pattern (SHSP) (b) Search Pattern of Hexagon Search.

### 13) Efficient Three Step Search

Xuan Jing et al. [94] proposed a modified version of three-step search algorithm for block matching motion estimation, which consists of unrestricted small diamond search pattern that is used to search the central area of defined search window and used in wide range of video applications like movies, sports etc. This algorithm initially defines a search window size and plots 13 points on that search window size. In step 1, outer 9 points and small diamond search pattern 4 points (Total 13 points) will be checked i.e. 4 points more than TSS and 4 points less than NTSS. If the minimum BDM point is in the center of search window then the search will be stopped/halts, otherwise goto step 2. In step 2, if minimum BDM point is one of the outer 8 points then TSS algorithm is used to search the point, otherwise goto step 3. In step 3, If minimum BDM point is one of the four points on the small diamond pattern, then consider this point as a center and checked another 3 points. This process is repeated until small diamond search pattern

reaches to search window boundary. In this algorithm, the number of check points required for Efficient Three Step Search (ETSS) are 13 for the best case and greater than or equals to 29 for the worst case because of unrestricted small diamond search pattern at central part of search window as shown in Fig. 15.

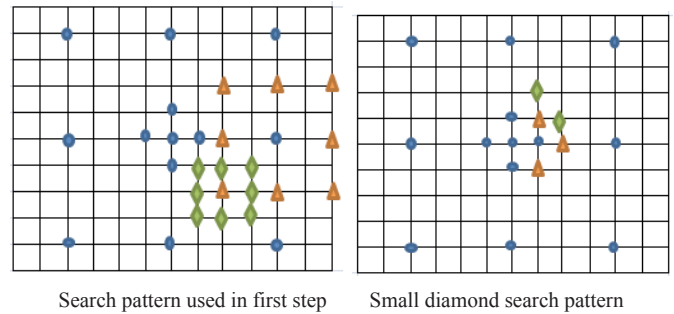


Fig. 15. Efficient Three Step Search.

### 14) New Cross Hexagonal Search

Kamel Belloulata et al. [88] introduces a new fast fractal cross hexagonal block matching motion estimation search algorithm consisting of two cross pattern search, i.e. a Small Cross Shape Pattern (SCSP) and Large Cross Shape Pattern (LCSP) as a few initial steps of search, and two cross hexagon search pattern, i.e. small and large hexagon search pattern as a subsequent steps of search as shown in Fig. 16, where SCSP: 1<sup>st</sup> step: ■, 2<sup>nd</sup> step: ◆, LCSP: 3<sup>rd</sup> step: ▲, LHSP: 4<sup>th</sup> step: ○ and ●, SHHP: 5<sup>th</sup> step: ■ Final MV. This algorithm initially starts with small cross shape pattern consisting of 5 points located at the center of the search window. If minimum BDM point is found at the center of the small cross shape pattern, then stop searching i.e. the number of checking points required for the new cross hexagonal search is 5 for the best case, which is better than all the techniques available for estimating a motion vector, otherwise consider a minimum BDM point as a center of newly formed small cross shape pattern. If minimum BDM point is found at the center of newly formed small shape pattern, then stop searching i.e. another best case solution is 8 checking points required for finding out best possible motion vector, otherwise check the other 3 unchecked points of large cross pattern and 2 unchecked points of the square center biased to show the best possible direction for the hexagonal search. A new large hexagonal shape pattern is formed by considering a center point as minimum BDM point found in small cross pattern search.

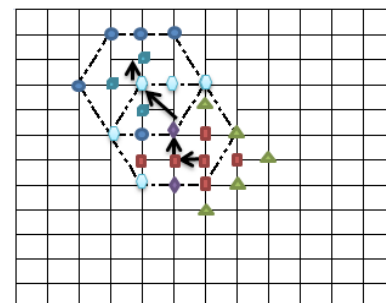
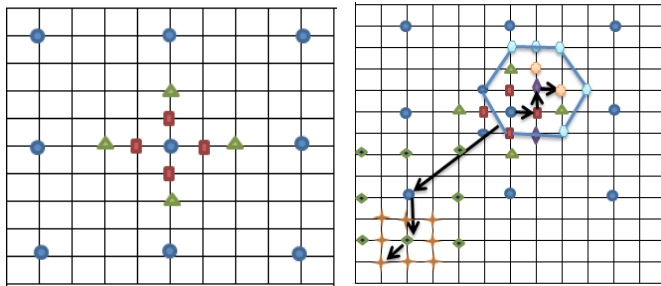


Fig. 16: Search Pattern in New Hexagon Search

If minimum point is found at the center of large hexagonal pattern, then large hexagonal pattern shifted /changed to small hexagonal pattern and find best motion vector in small hexagon shape pattern, otherwise again form a new large hexagon pattern, this formation of new large hexagon pattern is repeated until minimum BDM point is found at the center of large hexagonal pattern.

15) Modified Three Step Search

S. D. Kamble et al. [95] proposed another extended version of three-step search algorithm consisting of Two cross search pattern i.e. small and large cross search pattern and two cross hexagon search pattern i.e. large cross hexagon and small cross hexagon search pattern, which are used in the center part of search window to exploit central biased characteristic of MV in video sequences. Fig. 17 shows the search pattern used in modified three-step search. In the first step, total 9+4 points are checked out of 17 checking points. If minimum BDM point is found at the center of 9+4 points, then stop searching, otherwise go to the second step. If minimum BDM point is found at the outer part of the search window, then search process is same as TSS, otherwise go to the third step.



Search pattern used in first step Two Search pattern

Fig. 17: Search Pattern in Modified Three Step Search

If minimum BDM point is found at the 4 outer points of small cross search pattern, then search process is same as cross hexagon search. There is no restriction on searching in the center window part unless minimum BDM point is found at the center of large cross hexagon search pattern or large cross hexagon search pattern reaches to the outer boundary of the search window. This unrestricted search in the central part of the window increases the probability of finding a true motion vector within the center area of the window.

Some other approaches are also proposed by Acharjee et al. [96-98]. Smaller block size based motion estimation approach for video compression is proposed in [96]. Based on the movement in the video, the low and high motion zone approach exists in [97], while the scope of parallel processing is experimented in [98] for motion vector estimation. Generation of the motion vector and motion compensation prediction error plays a key role in video encoding process. Therefore, motion vector dominates the quality of reconstructed frames/video. Y.-G. Wu and G.-F. Huang [99] proposed motion vector generation using gray theory [100] proposed in 1982. Fig. 18 shows video compression process flow using motion compensation WFA encoder.

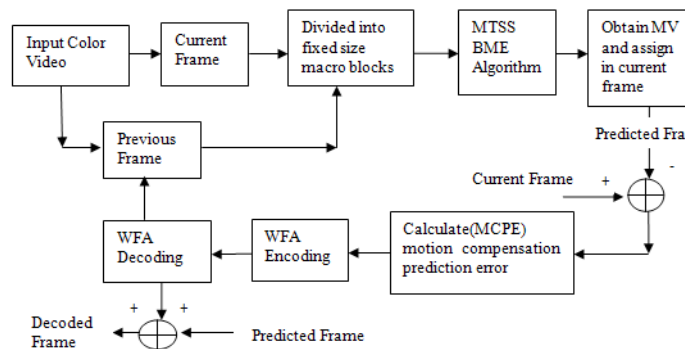


Fig. 18. Video Compression Process Flow.

VII. QUALITY MEASURE

Inter-frame and intra-frame coding are used to eliminate a large amount of temporal and spatial redundancy that exists in the video sequences and therefore, help in compressing them. The matching of the one current frame macro block and previous frame macro block is based on the output of matching criteria. The macro block that results in the minimum value is the one that matches the closest to current block with respect to the corresponding previous frame macro block. The popular matching criteria used for block matching motion estimation are mean of absolute difference (MAD), mean squared error (MSE) and sum of absolute difference (SAD) given by equation (8, 9 and 10) respectively.

$$MAD(i,j) = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (8)$$

$$MSE(i,j) = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (9)$$

$$SAD(i,j) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (10)$$

where  $N \times N$  is the row and column of the macro block,  $C_{ij}$  and  $R_{ij}$  are the pixels values compared in the current macro block and previous macro block, respectively. In Block matching algorithms, the size of macro block is the important parameter for motion estimation. Smaller macro block size results in more motion vectors and more macro blocks per frame. Therefore, quality of motion compensated prediction error (MCPE) is improved. Most video coding standards used a macro block of size  $16 \times 16$  and  $8 \times 8$ . The best/single motion vector is computed for each macro block in the reference frame. On the other hand, the total number of search points to find motion vector per frame is one of the key parameters in block matching motion estimation algorithm.

The performance of video coding is measured in terms of compression ratio, quality of the video, encoding time and decoding time. The compression ratio is given by equation (11).

$$\text{Compression ratio (CR)} = \frac{\text{Size of original Video}}{\text{Size of compressed file}} \quad (11)$$

Therefore, the compression ratio in percentage is computed from (11) and given by equation (12).

$$\text{Compression ratio (CR) in \%} = (1 - 1/CR) \times 100 \quad (12)$$

When measuring the quality of the compressed video, the peak signal-to-noise ratio is used. Sometimes mean squared error (mse) is also used, which is given by equation (13).

$$mse = \frac{1}{n} \sum_{i=1}^n (X_i - Y_i)^2 \quad (13)$$

From this, the PSNR for an 8-bit grayscale image is defined by equation (14)

$$PSNR(dB) = 10 \log_{10} \left( \frac{255^2}{mse} \right) \quad (14)$$

Where 255 is the maximum value an 8-bit pixel can assume.

VIII. DISCUSSION AND CONCLUSION

Based on the studied literature, it is found that there is a scope of improvement on the challenging issues of fractal coding like searching best domain block, domain pool size reduction, partitioning scheme, domain pool classification and use of parallel computing architecture for fractal compression. The major challenge in fractal coding is how to reduce the encoding time due to a large number of computations involved in fractal coding. From the studied literature on automata

theory based existing compression approaches, any image/sequence of image is represented by the finite automata. The automata theory based coding technique is similar to the fractal coding technique for searching the self similarity parts present in the image/sequence of image and regular expression notation is used to specify the address of each subimage. Our observations from the experimentation carried out by contributors/authors who have already contributed their work on automata theory based coding approach on the different existing databases, it achieves the high compression ratio, good image/video reconstruction quality, fast decoding and reduction in encoding time. The existing different block matching motion estimation approaches for finding out the motion vectors in a frame are widely accepted by video compression research community/society and efficiently used in video compression standards are- H.261 to H.263 and MPEG-1 to MPEG-4. Our observations from the studied literature on different block matching motion estimation algorithms are given in Table II. Table II shows the performance comparison for different existing approaches discussed in this paper based on number of search points per block and Table III shows comparative performance analysis based on the parameters i.e. step size, number of steps required and performance for small/large motions.

TABLE II  
PERFORMANCE COMPARISON BASED ON NUMBER OF SEARCH POINTS/ BLOCK

Search method	Search shape pattern	Best case	Average case	Worst case
TSS	Square	25	25	25
2DLS	Cross + Square	17	17	17
OSA	Line	13	13	13
CSA	Cross	17	17	17
BS	Square	17	17 to 33	33
NTSS	Square	17	17 to 33	33
FSS	Square	17	17 to 27	27
BBGDS	Square	09	09 to 34	34
DS	Diamond	13	13 to 30	30
CDS	Cross + Diamond	09	09 to 29	29
HS	Hexagonal	11	11 to 30	30
ETSS	Square + Diamond	13	13 to <=29	<= 29
CHS	Cross + Hexagonal	5/8	5/8 to <36	< 36
MTSS	Square + Hexagonal + Cross	13	13 to <29	< 29

TABLE III  
COMPARATIVE PERFORMANCE ANALYSIS ON PARAMETERS

Search method	Small motions	Large motions	Number of steps required
TSS	Not perform better	Perform better	3
2DLS	Not perform better	Perform better	3
OSA	Not perform better	Perform better	3
CSA	Not perform better	Perform better	Conditional
BS	Perform better	Not perform better	Boundary
NTSS	Perform better	Perform better	3
FSS	Not perform better	Perform better	4
BBGDS	Perform better	Not perform better	Boundary
DS	Perform better	Not perform better	Conditional
CDS	Perform better	Not perform better	Conditional
HS	Not Perform better	Perform better	Conditional
ETSS	Perform better	Perform better	3/ Boundary
CHS	Perform better	Perform better	Conditional
MTSS	Perform better	Perform better	3/Conditional

## IX. FUTURE SCOPE

There is a scope of improvement in developing a new shape pattern or modifying existing shape pattern approaches by combining two different algorithms i.e. developing a hybrid approach for finding out the motion vectors for fractal and other compression techniques with specific focus on reducing the encoding time and better reconstruction quality of video.

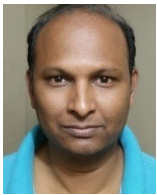
We can extend the block motion estimation approach in future by combing with the extended version of the finite automata for exploring the fractal video compression. We can also use these block matching motion estimation approaches for tracking a single/multiple object tracking applications to track the objects in a video.

## REFERENCES

- [1] M. F. Barnsley, *Fractals Everywhere*, Academic Press, 1988.
- [2] Acharjee, S., Dey, N., Biswas, D., Das, P., & Chaudhuri, S. S. (2012), "A novel Block Matching Algorithmic Approach with smaller block size for motion vector estimation in video compression", *12th IEEE International Conference on Intelligent Systems Design and Applications (ISDA)*, 2012, pp. 668-672.
- [3] Acharjee, S., Biswas, D., Dey, N., Maji, P., & Chaudhuri, S. S. (2013), "An efficient motion estimation algorithm using division mechanism of low and high motion zone", *IEEE International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, 2013, pp. 169-172.
- [4] Acharjee, S., Pal, G., Redha, T., Chakraborty, S., Chaudhuri, S. S., & Dey, N. (2014), "Motion vector estimation using parallel processing", *IEEE International Conference on Circuits, Communication, Control and Computing (I4C)*, 2014, pp. 231-236.
- [5] M. F. Barnsley and A. D. Sloan, "A better way to compress images", *Byte*, pp. 215-223, January 1988.
- [6] Y Fisher, "Fractal Image Compression", SIGGRAPH-1992 Course Notes.
- [7] R. C. Gonzalez, R. E. Woods, "Digital Image Processing", Second Edition, Pearson Education Asia, 2005.
- [8] L. P. Hurd, M. A. Gustavas, M. F. Barnsley. "Fractal Video Compression", AK Peters Ltd., 1993.
- [9] Thakur Nileshsingh V, Dr. O. G. Kakde, "Color Image Compression on Spiral Architecture using Optimized Domain Blocks in Fractal Coding", *IEEE 4th International Conference on Information Technology*, pp. 234-242, 2007.
- [10] C. S. Tong, and M. Wong, "Adaptive Approximate Nearest Neighbor Search for Fractal Image Compression", *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 605-615, June 2000.
- [11] S. K. Ghosh, J. Mukherjee, and P. P. Das, "Fractal Image Compression: A Randomized Approach", *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1013-1024, July 2004.
- [12] T. K. Truong, C. M. Kung, J. H. Jeng, and M. L. Hsieh, "Fast Fractal Image Compression using Spatial Correlation", *Chaos, Solitons & Fractals*, vol. 22, no. 5, pp. 1071-1076, Dec. 2004.
- [13] C. Wang, and J. Kao, "A Fast Encoding Algorithm for Fractal Image Compression", *IEICE Electronics Express*, vol. 1, no.12, pp. 352-357, 2004.
- [14] C. Fan, and P. Liu, "A Fast Matching Algorithm Based on Adaptive Classification Scheme", *5th IEEE International Conference on Cognitive Informatics, ICCI 2006*, vol. 1, pp. 541-546, 17-19 July 2006.
- [15] Q. Wang, D. Liang, Sheng Bi, "Fast Fractal Image Encoding Based on Correlation Information Feature", *3rd International Congress on Image and Signal Processing*, vol. 2, pp. 540-543, 2010.
- [16] S. K. Ghosh, J. Mukhopadhyay, V. M. Chowdary, and A. Jeyaram, "Relative Fractal Coding and Its Application in Satellite Image Compression", *Proceedings of ICVGIP, Ahmadabad, india*, pp. 469-474, Dec. 2002.
- [17] A. Conci, and F. R. Aquino, "Fractal Coding Based on Image Local Fractal Dimension", *Computational and Applied Mathematics*, vol. 24, no. 1, pp. 83-98, 2005.
- [18] J. Li, D. Yuan, Q. Xie, and C. Zhang, "Fractal Image Compression by Ant Colony Algorithm", *the 9th International Conference for Young Computer*

- Scientists, ICYCS 2008, pp. 1890-1894, 18-21 Nov. 2008.
- [19] H. Hartenstein, M. Ruhl, and D. Saupe, "Region-Based Fractal Image Compression", *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1171-1184, July 2000.
- [20] K. Belloulata, and J. Konrad, "Fractal Image Compression with Region Based Functionality", *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 351-362, April 2002.
- [21] H. Harenstein, and D. Saupe, "Lossless Acceleration of Fractal Image Encoding Via the Fast Fourier Transform", *Signal Processing: Image Communication*, vol. 16, no. 4, pp. 383-394, Nov. 2000.
- [22] R. Franco, and D. Mala, "Adaptive Image Partitioning for Fractal Coding Achieving Designated Rates Under A Complexity Constraint", *Proceedings of International Conference on Image Processing*, vol. 2, pp. 435-438, Oct. 2001.
- [23] Hai Wang, "Fast Image Fractal Compression with Graph-Based Image Segmentation Algorithm", *International Journal of Graphics*, vol. 1, no. 1, November-2010.
- [24] M. Hassaballah, M. M. Makky, and Y. B. Mahdy, "A Fast Fractal Image Compression Method Based Entropy", *Electronic Letters on Computer Vision and Image Analysis*, vol. 5, no. 1, pp. 30-40, 2005.
- [25] C. He, X. Xu, and J. Yang, "Fast Fractal Image Encoding using one-norm of normalized Block", *Chaos, Solitons & Fractals*, vol. 27, no. 5, pp. 1178-1186, March 2006.
- [26] N. Rowshanbin, S. Samavi, and S. Shirani, "Acceleration of Fractal Image Compression using Characteristic Vector Classification", *Canadian Conference on Electrical and Computer Engineering, CCECE 06*, pp. 2057-2060, May 2006.
- [27] F. Ce, "Fast Encoding Algorithm Based on Different Contrast Scales", *International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, pp. 1105-1108, 16-17 Aug. 2007.
- [28] J. Han, "Speeding Up Fractal Image Compression Based on Local Extreme Points", *Eighth ACIS International Conference on Software Engineering, Artificial intelligence, Networking, and Parallel/Distributed Computing, SNPD 2007*, vol. 3, pp. 732-737, 30 July-1 Aug. 2007.
- [29] C. Xing, Y. Ren, and X. Li, "A Hierarchical Classification Matching Scheme for Fractal Image Compression", *Congress on Image and Signal Processing, CISP'08*, pp. 283-286, 27-30 May 2008.
- [30] J. Han, "Fast Fractal Image Compression using Fuzzy Classification", *Fifth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD'08*, vol. 3, pp. 272-276, 18-20 Oct. 2008.
- [31] Z. Qin, H. Yu, and J. Ju, "Fractal Image Compression Based on Numbers of Hopping and Variance of Continuing Positive and Negative Pixels", *the 9th International Conference for Young Computer Scientists, ICYCS 2008*, pp. 2954-2958, 18-21 Nov. 2008.
- [32] R. Hashemian, and S. Marivada, "Improved Image Compression using Fractal Block Coding", *Proceedings of IEEE International Symposium on Micro-Nanomechatronics and Human Science*, vol. 2, pp. 544-547, Dec. 2003.
- [33] A. Kapoor, K. Arora, A. Jain, and G. P. Kapoor, "Stochastic Image Compression using Fractals", *Proceedings of the International Conference on Information Technology: Coding and Computing (Computers and Communications) (ITCC)*, pp. 574-579, April 2003.
- [34] S. K. Mitra, C. A. Murthy, M. K. Kundu, B. B. Bhattacharya, and T. Acharya, "Fractal Image Compression using Iterated Function System with Probabilities", *Proceedings of International Conference on Information Technology: Coding and Computing*, pp. 191-195, April 2001.
- [35] C. Gungor, and A. Ozturk, "A Hash Based Image Classification Technic and Application on Fractal Image Compression", *IEEE 15th Signal Processing and Communications Applications, SIU 2007*, pp. 1-4, 11-13 June 2007.
- [36] M. Salarian, and H. Hassanpour, "A New Fast no Search Fractal Image Compression in DCT Domain", *International Conference on Machine Vision, ICMV 2007*, pp. 62-66, 28-29 Dec. 2007.
- [37] K-W C. Eugene, and G-H Ong, "**A Two-Pass Improved Encoding Scheme for Fractal Image Compression**", *International Conference on Computer Graphics, Imaging and Visualization*, pp. 214-219, 26-28 July 2006.
- [38] V. R. Prasad, Vaddella, R. Babu, and Inampudi, "Adaptive Gray Level Difference To Speed Up Fractal Image Compression", *International Conference on Signal Processing, Communications and Networking, ICSCN '07*, pp. 253-258, 22-24 Feb. 2007.
- [39] Z. Peng, T. C. S. Tian, H. Zhao, and F. Meng, "Fractal Image Coding Based on High Order Spectrum using nonparametric Estimation", *International Conference on Communications, Circuits and Systems, ICCAS 2007*, pp. 812-815, 11-13 July 2007.
- [40] D. J. Duh, J. H. Jeng, and S. Y. Chen, "Fractal Image Compression with Predicted Dihedral Transformation", *12th IEEE Symposium on Computers and Communications, ISCC 2007*, pp. 661-666, 1-4 July 2007.
- [41] X. Liu, Y. Zhang, and D. Li, "Implementation of A Quick Fractal Image Compression Algorithm and Its Application in Digital Certificate Solution Scheme", *IEEE International Conference on Automation and Logistics*, pp. 441-445, 18-21 Aug. 2007.
- [42] G. N. Melnikov, "the Fractal Method of the Image Coding", *Siberian Conference on Control and Communications, SIBCON 07*, pp.153-157, 20-21 April 2007.
- [43] Riccardo Distasi, Michele Nappi, and Daniel Riccio, "A Range/Domain Approximation Error-Based Approach for Fractal Image Compression", *IEEE Transactions on image processing*, VOL. 15, NO. 1, January 2006.
- [44] E. W. Jacobs, Y. Fisher, and R. D. Boss, "Image Compression: A Study of Iterated Transform Method", *Signal Processing*, vol. 29, no. 3, pp. 251-263, Dec. 1992.
- [45] L. Moltredo, M. Nappi, D. Vitulano, and S. Vitulano, "Color Image Coding Combining Linear Prediction and Iterated Function Systems", *Signal Processing*, vol. 63, no. 2, pp. 157-162, Dec. 1997.
- [46] B. Hurtgen, P. Mols, and S. F. Simon, "Fractal Transformation Coding of Color Images", *Visual Communication and Image Processing, SPIE Proceedings*, pp. 1683-1691, 1994.
- [47] Y. Zhang and L. M. Po, "Fractal Color Image Compression using Vector Distortion Measure", *Proceedings of IEEE International Conference on Image Processing*, vol. 3, pp. 276-279, 23-26 October 1995.
- [48] I. M. Danciu, and J. C. Hart, "Fractal Color Compression in the  $L^*a^*b^*$  Uniform Color Space", *Data Compression Conference*, pp. 540, March 1998.
- [49] Z. Li, L. Zhao, and N. Y. Soma, "Fractal Color Image Compression", *Proceedings of the 13th Brazilian Symposium on Computer Graphics and Image Processing*, pp. 185-192, 2000.
- [50] D. D. Giusto, M. Murrioni, and G. Soro, "Fractal Coding of Color Images using Earth Mover's Distance", *Proceedings of the 2nd International conference on Mobile Multimedia Communications, Mobimedia'06, ACM*, vol. 324, pp. 1-5, 18-20 Sept. 2006.
- [51] Thakur N. V. and Kakde O. G. (2007b) „Color image compression with modified fractal coding on spiral architecture”, *Journal of multimedia*, vol. 2(4), p.p.55-66.
- [52] K. Culik and J. Kari, "Inference algorithms for WFA and image compression". In Y. Fisher, editor, *Fractal Image Compression*, chapter 13, pages 243-258. Springer-Verlag, 1995.
- [53] J. Kari and P. Franti, "Arithmetic coding of weighted finite automata. *Theoretical Informatics and Applications*", 28(3-4):343-360, 1994.
- [54] U. Hafner, Lehrstuhl für Inf., Würzburg Univ., Germany "Refining Image Compression with Weighted Finite Automata", presented at the IEEE Data compression Conference, March 1996, pp. 359-368.
- [55] F. Katritzke, "Refinements of Data Compression Using Weighted Finite Automata", Ph.D. dissertation, graph. Darst. - Siegen, Univ., Diss., 2001.
- [56] F. Katritzke, W. Merzenich, M. Thomas, "Enhancements of partitioning techniques for image compression using weighted finite automata", Elsevier, 2003.
- [57] Y. Fisher, T. P. Shen and D. Rogovin, "Fractal (Self-VQ) encoding of video sequences," *VCIIP*, vol. 2308, pp. 1359-1370, 1994.
- [58] C. S. Kim, R. C. Kim and S. U. Lee, "Fractal coding of video sequence using circular prediction mapping and noncontractive interframe mapping", *IEEE Transaction on Image Processing*, vol. 7, no. 4, pp. 601-605, 1998.
- [59] D. T. Hoang, P.M. Long and J. S. Vitter. Explicit bit minimization for motion-compensated video coding. In J. A. Storer and M. Cohn editors, *Proc. of the Data Compression Conference*, Pages 175-184, 1994.
- [60] Ullrich Hafner, Stefan Frank, Michael Unger, Jurgen Albert. Hybrid Weighted Finite Automata for Image and Video Compression. Report 160 (revised) March 1997.

- [61] Paul Bao, Xiaohu Ma. Video Coding Based on Bitplane and GFA Modeling. 0-7803-9584-0/06, 2006 IEEE. Pp. 109-113.
- [62] Paul Bao and Xiaohu Ma, School of Computer Engineering, Nanyang Technological University, Very Low Bit rate Video Compression based on GFA Modeling. 2004, IEEE, International Conference on Multimedia and Expo (ICME).
- [63] Gary J. Sullivan and Thomas Wiegand, Video Compression- From Concepts to the H.264/AVC Standard. 0018-9219/\$20.00 © 2005 IEEE. Proceedings of the IEEE, VOL. 93, NO. 1, JANUARY 2005.
- [64] Karel Culik II, Vladimir Valenta. Finite Automata Based Compression of Bi-level and Simple Color Images. Utah: Data Compression Conference, 1996.
- [65] Xiaohu Ma, Huanqin Chen. Image Compression Method Based on Generalized Finite Automata. ICALIP, IEEE, 2008, pp. 1688-1692.
- [66] Kamble, S. D., Thakur, N.V., Malik, L. G. and Bajaj, P. R. (2015b) „Color video compression based on fractal coding using quad-tree weighted finite automata”, Information system design and intelligent application, Proceedings of Second International Conference INDIA 2015, vol.2, advances in intelligent system and computing, Springer india, vol. 340, pp-649-658.
- [67] Thakur N. V. and Kakde O. G. (2006a) „Fractal color image compression on a pseudo spiral architecture”, IEEE International Conference on cybernetics and intelligent systems, p.p.1-6.
- [68] Thakur N. V. and Kakde O. G. (2006b) „A novel compression technique for color image database”, IEEE International Conference on advanced computing and communications, p.p. 240–243.
- [69] J. Domaszewicz and V. A. Vaishampayan, “Graph-theoretical analysis of the fractal transform”, in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP’95, Apr. 1995, vol. IV, pp. 2559-2562.
- [70] M. Wang, R. Liu, and C. H. Liu, “Adaptive partition and hybrid method in fractal video compression”, Computers & Mathematics with Applications, vol. 51, no. 11, pp.1715-1726, 2006.
- [71] T. Ochotta and D. Saupe, “Edge-based partition coding for fractal image compression”, The Arabian Journal for Science and Engineering, Special Issue on Fractal and Wavelet Methods, vol. 29(2C), 2004.
- [72] K. Belloulata and J. Konrad, “Fractal image compression with region-based functionality”, IEEE Transaction on Image Processing, Vol. 11, no. 4, pp. 351-362, Apr. 2002.
- [73] K. Belloulata, “Fast fractal coding of subbands using a noniterative block clustering”, Journal of Visual Communication and Image Representation, N°16, pp. 55-67, Feb. 2005.
- [74] K. Belloulata and S. Zhu, “A New Object-Based System for Fractal Video Sequences Compression”, In the Data Compression Conference, DCC’08, Utah, USA, March 2008, pp. 508 510.
- [75] K. Belloulata and S. Zhu, “A New Object-Based Fractal Stereo CODEC with quadtree-based disparity or motion compensation”, in the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP’06, Toulouse, France, 14-19 may 2006, Vol. II, pp. 481-484.
- [76] S. Zhu and K. Belloulata, “A novel object-based fractal stereo video codec”, in the IEEE Int. Conf. Image Processing, ICIP’05, Genova, Italy, 22-24 Sept. 2005, Vol. I, pp. 805-808.
- [77] S. Zhu, J. Tian, X. Shen, K. Belloulata, “A new cross-diamond search algorithm for fast block motion estimation”, in the IEEE Int. Conf. Image Processing, ICIP’09, Cairo, Egypt, 07-11 Nov. 2009, Vol. I, pp. 1581-1584.
- [78] S. Zhu, J. Tian, X. Shen, K. Belloulata, “A Novel Cross- Hexagon Search Algorithm Based on Motion Vector Field Prediction”, in the IEEE Int. Symposium on Industrial Electronics, ISIE’09, Seoul, Korea, July 2009, pp. 1870-1874.
- [79] S. Zhu, Y. Hou, Z. Wang and K. Belloulata, “A novel fractal video coding algorithm using fast block matching motion estimation technology”, in the International Conference on Computer Application and System Modeling, ICCASM’10, Taiyuan, China, Oct. 2010, Vol. 8, pp.360-364.
- [80] Information Technology-Coding of Audio Visual Objects-Part 2: Visual, ISO/IEC 14 469-2 (MPEG-4 Visual), 1999.
- [81] R. Li, B. Zeng, and M. L. Liou, “A new three-step search algorithm for block motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no.4, pp. 438-443, 1994.
- [82] L.-M. Po and W.-C. Ma, “A novel four-step search algorithm for fast block motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 313–317, 1996.
- [83] L. K. Liu and E. Feig, “A block based gradient descent search algorithm for block motion estimation in video coding”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 4 , pp. 419-422, 1996.
- [84] S. Zhu and K. K. Ma, “A new diamond search algorithm for fast block-matching motion estimation”, IEEE Transactions on Image Processing, vol. 9, no. 2, pp. 287-290, 2000.
- [85] C. H. Cheung, and L. M. Po, “A novel cross-diamond search algorithm for fast block motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 12, pp. 1168-1177, 2002.
- [86] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, “A novel unrestricted center-biased diamond search algorithm for block motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 4, pp. 369-377, 1998.
- [87] H. M. Chen, P. H. Chen, K. L. Yeh, W. H. Fang, and M. C. Shie and F. Lai, “Center of Mass-Based Adaptive Fast Block Motion Estimation”, EURASIP Journal on Image and Video Processing, vol. 2007, Article ID 65242, 11 pages
- [88] Ce Zhu, Xiao Lin, and Lap-Pui Chau, “Hexagon-based search pattern for fast block motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, pp. 349-355, May 2002.
- [89] C. H. Cheung and L.M Po, “Normalized partial distortion search algorithm for block motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, pp. 417-422, March 2000.
- [90] Li Hong-ye Liu Ming-jun “Cross-Hexagon-based motion estimation algorithm using motion vector adaptive search technique” International Conference on Wireless Communications & Signal Processing, 2009.
- [91] Kamel Belloulata, Shiping Zhu, Zaikuo Wang, “A Fast Fractal Video Coding Algorithm Using Cross-Hexagon Search for Block Motion Estimation”, International Scholarly Research Network Signal Processing, volume 2011, Article ID 386128, 2011.
- [92] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion compensated interframe coding for video conferencing,” in Proc. National Telecommunications Conf., New Orleans, LA, pp. G5.3.1 –G5.3.5., 1981.
- [93] J. R. Jain and A. K. Jain, “Displacement measurement and its application in interframe image coding” IEEE Transactions on Communications, vol. 29, pp. 1799-1808, Dec. 1981.
- [94] A. Puri, H. M. Hang and D. L. Schilling, “An efficient block matching algorithm for motion compensated coding,” Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc., pp. 1063-1066, 1987.
- [95] M. Ghanbari, “The cross search algorithm for motion estimation,” IEEE Trans. Commun., Vol. COM-38, pp. 950-953, Jul. 1990.
- [96] Toshiyuki Urabe, Hassan Afzal , Grace Ho, Pramod Pancha and Mogda El Zarki , “MPEG Tool Version 1.03.Readme File”, Department of Electrical Engineering, Univesity of Pennsylvania, Philadelphia.
- [97] Xuan Jing and Chau Lap-Pui, “An efficient three-step search algorithm for block motion estimation,” IEEE transactions on multimedia, Vol. 6, No.3, pp. 435-438, 2004.
- [98] S. D. Kamble, N. V. Thakur, L.G. Malik and P. R. Bajaj, “Fractal Video Coding Using Modified Three-step Search Algorithm for Block-matching Motion Estimation”, Computational Vision and Robotics Proceedings of International Conference on Computer Vision and Robotics, ICCVR’14, Advances in Intelligent Systems and Computing ,Vol. 332, pp 151-162 Springer-India, 2015.
- [99] Y. G. Wu and G. F. Huang, “Motion vector generation for video coding by gray prediction”, IET Comput. Vis., 2011, Vol. 5, Iss. 1, pp. 14–22.
- [100] Deng, J.: ‘Control problems of grey system’, Syst. Control Lett., 1982, 5, pp. 288–294.



**Shailesh D. Kamble** received Bachelor of Engineering degree in Computer Technology from Yeshwantrao Chavan College of Engineering, Nagpur, India under the Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur, India. He received Master of Engineering degree from Prof Ram Meghe Institute of Technology and Research, formerly known as College of Engineering, Badnera under Sant Gadge Baba Amravati University, Amravati, India. Presently, he is Ph.D. candidate in department of Computer Science and Engineering from G.H. Rasoni College of Engineering (An Autonomous Institution Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University), Nagpur, India. He is working as the Assistant Professor in Department of Computer Science and Engineering at Yeshwantrao Chavan College of Engineering, Nagpur, India. He is having over 15 years of teaching and research experience. His current research interests include image processing, video processing and language processing. He is the author or co-author of more than 20 scientific publications in International Journal, International Conferences, and National Conferences. He is the life member of ISTE, India, IE, India.



**Nileshsingh V. Thakur** received Bachelor of Engineering degree in Computer Science Engineering from Government College of Engineering, Amravati, India and Master of Engineering degree from College of Engineering, Badnera under Amravati University and Sant Gadge Baba Amravati University in 1992 and 2005 respectively. He received Ph.D. degree in Computer Science Engineering under Department of Computer Science Engineering from

Visvesvaraya National Institute of Technology, Nagpur, India on 1st February, 2010. His research interest includes image and video processing, sensor network, computer vision, pattern recognition, artificial neural network and evolutionary approaches. He is having over 24 years of teaching and research experience. Presently, he is Professor and Head in Department of Computer Science and Engineering and Dean, PG Studies at Prof Ram Meghe College of Engineering and Management, Badnera- Amravati, Maharashtra, India. He is the author or co-author of more than 70 scientific publications in refereed International Journal, International Conferences, National Journal, and National Conferences. He is a member of editorial board of over eight International journals; also, he is the life member of ISTE, India, IAENG and IAEME. He also worked as the reviewer for refereed international journals and conferences.



**Preeti R. Bajaj** is an Electronics Engineer Graduated in 1991, Post graduate in 1998 and awarded doctorate in Electronics Engg in 2004. Having 25 year of experience, currently she is Director of G.H. Rasoni College of Engineering, Nagpur. Her research interest includes Intelligent Transportation System, Soft Computing, Hybrid Intelligent Systems & Applications of Fuzzy logic in ITS. Her professional society affiliation includes Fellow-

Institute of Engineers, Fellow IETE, Senior Member- IEEE, LM-ISTE, and LM-CSI, member ACM. She is presently Vice chair students activities- India council for IEEE. She is the first Indian to be selected as Technical Committee Chair on System Man and Cybernetics Society of the IEEE. She had been consultant for NHAI, Government of India. Six Doctoral students have been awarded PhD & 15 have done masters under her. She has authored & co-authored 100 plus reviewed Publications. She has published three book chapters. She is founder General Chair of ICETET series of IEEE Conferences. Under her leadership GHRCE has been awarded Autonomy, NBA accreditation to 21 programs, NAAC with A Grade, TEQIP World Bank project 1 and 2 leading to transform a private Engineering institution into India's premier, Engineering and Research Institute.