

Fractal Video Compression using Block Matching Motion Estimation - A Study

Rakhi Ashok Aswani, Prof.S.D.Kamble

Computer Technology. Yeshwantrao Chavan College of Engg. & Tech. Nagpur, India
Computer Technology. Yeshwantrao Chavan College of Engg. & Tech. Nagpur, India

Abstract: The large usage of multimedia applications on internet and mobiles has increased the demand of compressed data, as to reduce the bandwidth requirement and transmission time. So block based method for video compression is getting more importance due to its effectiveness and easy implementation. Fractal video compression follows the property of self-similarity. The weighted finite automata exploits the property of self-similarity in a single image or sequence of images. In block matching motion estimation various algorithm has been proposed having different search pattern and strategies. The paper presents the review on different methods for fractal video compression.

Index Terms: fractal video compression, motion detection, motion estimation, partitioning scheme, key issues, performance measures.

I. INTRODUCTION

In today's digital world bandwidth requirement is most concerning factor for many multimedia applications. And efficient coding of video data in video file formats becomes an important process for many multimedia applications and requires a very high compression ratio. So here the compression plays the important role to save the bandwidth as well as the time to access the data. While compressing an image or video the most important thing is finding the similarity among frames or pixels. The fractals uses the concept of self similarity. Fractal image compression reduces the redundancy of images by using self-similarity properties and seems to be a favorable method for the image compression due to its advantages of high compression ratio, fast decompression and resolution independence[42].

The similarity can be interpixel or intrapixel similarity. The interpixel similarity means similarity among two consecutive frames. While intrapixel similarity deals with the similarity among neighboring pixels. The method is best suited for textures and natural images, on the fact that parts of an image often resembles to other parts of the same image. Fractal algorithms converts these parts into mathematical data called "fractal codes" which are used to recreate the encoded image. There are many proposed method on fractal image compression. Fractal Video Compression is performed in the same way as Fractal Image Compression is achieved. The major difference between them is that, video compression takes advantage of the similarity between successive frames and hence we obtain higher compression rates compared with image compression[16].

Fisher classified the image blocks (range block and domain block)[43]. An image block was divided into four quadrants. For each of the quadrant, average and the variance were computed. The range block found matches only in the domain pool with the same class. This reduces the search space efficiently [44]. However it needs a large amount of computations in the classifying process and some range blocks may not find the matching blocks in the same class in the domain pool [45].

The fig.1 shows different frame sequences with inter-frame mapping (shown in hard arrow) and domain-range mapping (shown in dashed arrow) in two consecutive frames.

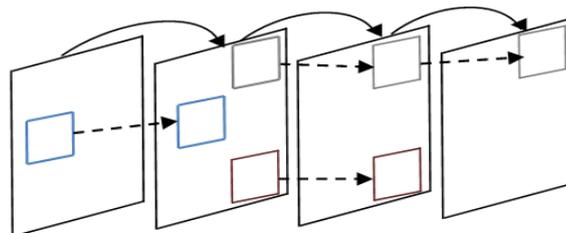


Figure1 Inter-frame and Domain-Range mapping

II. FRACTAL VIDEO COMPRESSION

Fractal image compression has become an accepted method for compressing still images. The main idea behind fractal compression is that it exploits the self-similarity present in images. The fractal video compression

is a lossy compression technique. It works well for videos with steady background instead of continuously changing background. It gives best results when used for natural images, but also gives worst results with videos containing textual information. But the fractal compression can achieve very high compression ratio 10000:1, which attracts researchers to work in this field.

The fractal compression can be cube based or frame based. In *cube-based compression* a large sequence is partitioned into groups of frames (GOFs). Each of which is partitioned into non-overlapped cubes. The compression code is computed and stored for each cube. This method has a high computing complexity which makes it difficult to implement due to the limit of computing power, random access memory, etc. In the *frame based compression*, the compression codes are computed for each frame and stored, and intra-frame/inter-frame self-similarity can be used. The former method can be used to obtain high quality images, but in this case, the compression ratio is relatively low; the later method may be used to obtain a high compression ratio in such a way that the current frame is made to relate to the previous frame. Then, a hybrid compression algorithm is proposed. It consists of the frame-based algorithms and the cube-based compression algorithm[16].

III. WEIGHTED FINITE AUTOMATA

The weighted finite automata (WFA) method attempts to represent a subimage as a weighted sum of other subimages to accomplish compression. WFA starts with an image to be processed. It locates subimages that are identical or very similar to the entire image or to other subimages, and subsequently constructs a graph that reflects the relationship between these subimages with the entire image. The various components of the graph are compressed and the final product is a much more compactly represented domain.

An important issue involved in fractal-based image compression is the partitioning of an image into subimages or blocks for encoding. There are various approaches to the use of different block shapes. One such commonly used approach is based on quadtree partitioning. It has also been shown that a partitioning scheme based on rectangular blocks is simple yet offers high quality of decomposed images[31].

Fig. 2 depicts a linear WFA inferred for the target image x in the double lined square box. Each state is represented by its state image and accompanied by its average intensity in the corresponding arrowed bracket. A typical label “ $a(w)$ ” refers to the transition on the symbol $a \in Q = \{0; 1; 2; 3\}$ with w denoting the weight.

An extended WFA is a preferred WFA to represent a target image because it has less number of states, thus requiring typically less space. The extended WFA will consist of (i) a finite set of states, $S = \{1; 2; \dots; n\}$; (ii) a finite alphabet Q ; (iii) a target state; (iv) a final distribution (v) a set of weight functions. The pump states(q states) refers to some preselected base images, and these base images at any resolutions may be generated, or pumped out, independent of the other state images[36].

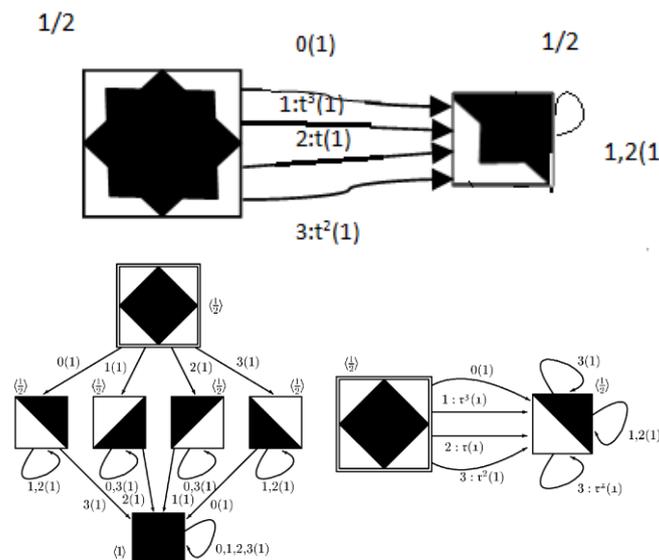


Figure 2. Linear WFA and Extended WFA

IV. PARTITIONING SCHEME

In block motion compensated video coding, first step is dividing the image frames into square blocks as-

A. Fixed Size Partitioning

Block Motion Compression using fixed sized blocks has disadvantages that large blocks may fail to match the slight motion in a sequence, specially along the moving edges, while small blocks requires more overhead also. Mostly, the objects in motion have different sizes and shapes. In order to overcome the limitations of the traditional fixed-size block motion compression(FSBMC), other techniques were proposed.

B. Variable Size Partitioning

Achieving, both good video compression ratio and image quality is a conflicting requirement, with *fixed-size block matching* (FSBM). The problem can be overcome by variable size block matching motion estimation technique (VSBME). In this, smaller blocks can be used to describe complex motion while larger blocks can be used where the image content is stationary or undergoing uniform motion, also the no. of blocks and bit rate are fixed

C. Quad-Tree Partitioning

The quadtree representation of an image gives a hierarchical view in which the root node represents the whole image[22]. In a two-level quadtree structure, one might begin with a regular partitioning of 8 x 8 blocks. For each range block, the domain pool is searched for the best match. If the accuracy of the matched falls within a least tolerance, it is accepted. If it is not, the range block is subdivided and a search is initiated for each sub-block[47].

So, the quadtree is a tree of degree four, with each inner node having four children. Its leaves have to fulfill a certain simplicity criterion, e.g., all pixels in the pertaining quadtree have equal gray values. The quadtree generation algorithm is illustrated in Fig. 3. The nodes are traversed in the order represented by the tree on the right. The leaves store the gray values “B” (black) and “W” (white). The inner nodes are labeled with “G” (gray) to indicate that they have to be partitioned further. A quadtree is suited for image compression because large areas can often be stored in a single node. The quadtree can also be used for lossy image coding if the recursive division is stopped if a special “homogeneity property” is fulfilled by the addressed segment. The associated node of the quadtree could store a gray value which minimizes a given distortion function, like the mean value for a quadratic error measure[35].

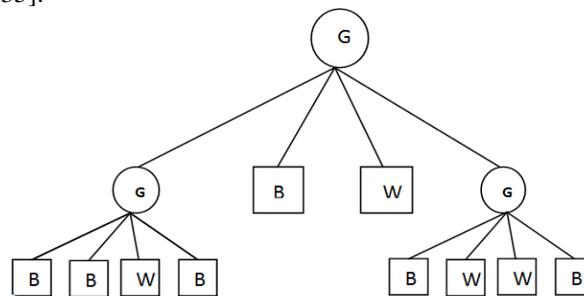


Figure 3 Quadtree representation of an image

The Quad-tree Decomposition(QD) fractal encoding scheme serves as reference point for various fractal schemes. Although, the analysis indicates that in QD approach involves many computations are performed which are not required. So instead of QD approach Quad-tree Recomposition(QR) is introduced which reduces the encoding time by removing not required calculations, without affecting the fidelity of reconstructed image[23].

V. PROCEDURES FOR FRACTAL COMPRESSION

The next step is a three step procedure, consisting of a)Motion Detection, b)Motion Estimation and c)Motion Compensation.

A. Motion Detection-

Motion detection is used for classifying blocks as moving or *non-moving* based on a predefined distance or similarity measure. This similarity measure is usually done by MSE, SAD,SSD etc.

B. Motion Estimation

In motion estimation, for each block in the frame, a search is made in an reference frame of the sequence over a predefined area of the image. The best matching block is searched, the position which has minimum distortion measure. The Minimum Block Distortion Measure(BDM) is the motion vector. The motion vector finds the similarity in blocks in domain frame and reference frame.

The most straightforward approach is exhaustive search. This is computationally demanding, but also it is algorithmically simple, and relatively easy to implement in hardware. The output of the motion-estimation algorithm known as Block Matching Algorithms(BMA), comprises the motion vector for each block, and the pixel value differences between the blocks in the current frame and the “matched” blocks in the reference frame. The difference signal also known as the motion compensation error, or simply block error[18].

All the methods are proposed keeping any one or more of the three keys aimed that 1.trying to reduce computational complexity 2.To represent the true motion (proving good quality) And 3.To reduce the bit rate(high compression ratio)[18].

VI. RELATED WORK

Block-matching motion estimation(BME) 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 encoding process[46].

The BMAs (Block Matching Algorithms) can perform the search in full search or fast search manner. The full search method is very simple and having less complexity. It searches in the entire search area to find the best matching block. The fact that the global optimal solution can be achieved using this kind of method proves the advantage of FS algorithm. However, due to its high computational complexity, it is not considered as a good choice for real time video coding implementation. Thus, many fast BMAs have been proposed and developed, for example, three step search (TSS), new three step search (NTSS),Improved three step search(ITSS), four step search (4SS), diamond search (DS), cross diamond search (CDS), extended diamond search(EDS), new cross diamond search (NCDS), cross diamond hexagonal search (CDHS) and hexagon-diamond search (HDS), to reduce the computational cost as much as possible while at the same time making ME accuracy degradation as least as possible. Different search patterns and search strategies are employed in these fast BMAs to find the global optimal solution with reduced number of search points as compared with the FS algorithm.

Fractal image compression is originated from Barnsley’s research. Barnsley [1] applied fractal image compression based on IFS system to computer graphics, but the approach requires manual intervention. Subsequently, Jacquin [2] proposed a new fractal image compression method based on image block, and the method can conduct automatically without manual intervention. Therefore, Jacquin’s method has become a typical representation, fractal image compression has become practical since then. Currently, fractal image compression has got extensive attention by the research community because of its novel ideas, high compression ratio and resolution independent characteristics.

Although fractal compression has many advantages, but in the process of fractal encoding, it needs a lot of computing, and this results in a long encoding time. Currently, fractal compression’s research focuses on how to shorten the encoding time.

Shiping Zhu [3] proposed an adaptive threshold quad tree fractal compression approach, compared with fixed square segmentation compression approach, it has a greater flexibility. It divides the image block which has high details into smaller sub-block, and for the image block with low details, divides them into larger sub-block. By doing so, it can reduce the number of image blocks which need to match and shorten the encoding time.

Hai Wang [4] has used concept of content based image segmentation. In the fractal encoding process, most blocks which come from different image’s contents have different textures, so blocks come from different image contents can’t be the best-match block for each other in most cases, but for the blocks come from the same image content, they can easily match each other well. It combines graph-based image segmentation algorithm and adaptive threshold quad tree fractal compression approach together, and propose a new fractal image compression approach. Graph-based image segmentation approach to segment image at an appropriate segmentation scale, separate the original image into many different logic areas according to image content, and construct the corresponding search space for each logic area. Then encode each logic area using adaptive threshold quad tree approach. When the encoding for all the logic areas in the whole image is finished, the whole image’s encoding is completed. When encoding with adaptive threshold quad tree approach, for each R block, it doesn't requires to find its best-match block within the entire image, but find it within the local search space in which R block located. Thus, use local optimum to replace global optimum.

Now we review the recent studies reported on block based motion compensation techniques. Among the BMA’s, Three step search(TSS) became the most widely used technique mainly because of its faster estimation. Koga [5] proposed TSS, which consists of three evaluation steps—each step contains nine uniformly spaced search points which get closer after every step. The best candidate search point in the previous step becomes the center of the current step. Hence, TSS requires (9+8+8=25) a fixed search points per block, which leads to a speedup ratio of 9 over the FS when W=7. The main drawback of TSS is the relatively large search pattern in the first step which renders it inefficient for finding blocks with small motions. This goes against the fact that most real-world sequences have a centrally biased motion vector distribution.

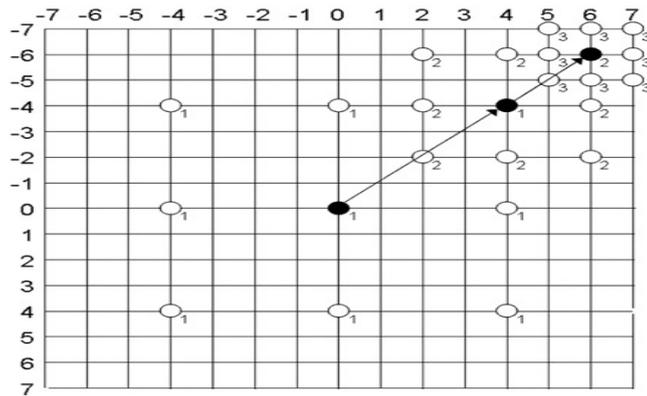


Figure 4.Steps involved in Three - step search

In order to exploit the characteristics of the center-biased motion vector distribution, a new three-step search (NTSS) algorithm was introduced by R. Li, B. Zeng [6]. It employs a center-biased search pattern in the first step by adding a smaller central eight-point pattern to that of the TSS. As a result, the worst case scenario of the NTSS will require $(25+8)=33$ block matches. However, the NTSS also allows termination of the search after the first or second step. Using this technique, only 17 search points are needed for stationary blocks, and either 20 or 22 search points for quasi-stationary blocks. The speed of NTSS is within 18% of TSS, but it gives almost consistently better motion estimates. However, the computational requirement of NTSS may be higher than the TSS for sequences which have a lot of large motion vectors.

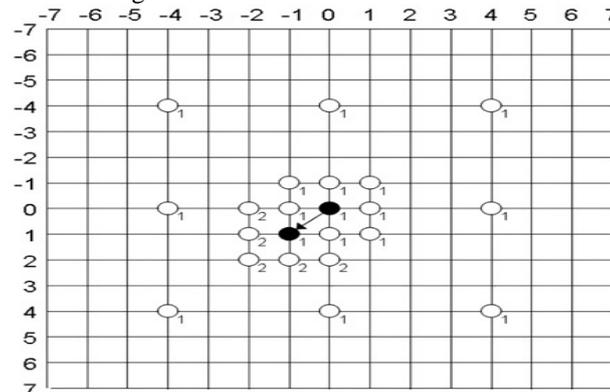


Figure 5.Steps in New Three-Step Search

Donglai Xu [7] proposed the Improved three step search(ITSS) employs a smaller number of search points than TSS in order to speed up block matching. The first step of the nine checking points on a 5x5 window in contrast to the 9x9 window of TSS. The centre of the search window is then shifted to the point with minimum BDM. The search window size of the next two steps depends on the location of the minimum BDM point. If the minimum lies at the centre of the search window, the search will go to the final step(3) with a 3x3 search window. Otherwise, the search window size is maintained at 5x5. In the final step, the search window is reduced to 3x3 and the search stops at this small search window.

L. M. Po [8] proposed 4SS produces better performance than the 3SS and has similar performance as compared with the N3SS. It has better speed up on both the worst case and average-case computational requirements of NTSS. It also exploits the center-biased motion vector distribution characteristic by utilizing a nine-point search pattern on a 5x5 grid in the first step instead of a 9x9 grid as in the TSS. As a result of starting with a smaller search grid pattern, 4SS requires four search steps as compared to only three steps in both the TSS and NTSS for the same search window of $W=7$. Nevertheless, simulation results show that the total number of candidate search points in 4SS actually ranges from the best case of 17 to the worst case of 27 points. The 4SS gives a speedup of six block matches for the worst case, and an average of two block matches less than the NTSS. More importantly, 4SS still manages to maintain motion estimation performance comparable to the NTSS, which in turn is better than the TSS.

Ming-gang Liu [9] proposed the Diamond search(DS), instead of typical rectangular shape used in TSS, NTSS and 4SS algorithms, the DS algorithm uses a diamond shape which tries to behave as an ideal circle-shaped coverage for considering all possible directions of an investigating MV. DS algorithm uses a large diamond search pattern (LDSP) and a small diamond search pattern (SDSP). DS also utilizes the overlapping of checking points between successive steps to reduce the computational complexity further. In terms of both average number of search points per motion vector and average mean square error (MSE) per pixel. DS is the best

of DS, 4SS and NTSS. Additionally, for a stationary block, DS still needs 13 checking points to get its final motion vector.

Chun-Ho Cheung [10] Cross-Diamond Search(CDS) Algorithm is proposed. It is employed by introducing a cross-shaped search pattern (CSP) as the initial step, instead of the diamond-shaped in the DS algorithm. Experimental simulations show that it can achieve fewer search points over DS and can maintain similar or even smaller distortion error.

Queen Mary [11] Hexagon-diamond search(HDS), instead of LDSP shape used in the initial step of DS algorithm, the proposed HDS algorithm employs a hexagon shape for its first and second steps before using the same SDSP shape, used in DS algorithm, in its final search step. The hexagon is proposed because most of the block displacements of real-world video sequences are mainly in horizontal and vertical directions. HDS outperformed FS, TSS, NTSS, 4SS and DS algorithms in terms of search points required in determining the MV of the block.

Kamel Belloulataa [12] proposed the new cross-hexagon search(NCHEXS) consists of two patterns: cross-based and hexagon-based. As the motion vectors distribution possesses cross-center biased characteristics in the central 4x4 pixels area, two cross-shaped patterns: small-cross-shaped (SCSP) and large-cross-shaped pattern (LCSP) are created. Also large and small hexagon patterns are created. The large hexagon pattern consists of not only the 7 check points in classic large hexagon pattern, but also the two edge points. So it requires total 9 check points in best case. For stationary blocks, HEXS and current cross-diamond algorithm takes 13 and 9 search points respectively, while NCHEXS takes 5 search points. For quasi-stationary blocks, HEXS and current cross-diamond algorithm takes 13 and 11 search points respectively, while NCHEXS takes 7 search points.

Chun-Ho Cheung [13] designed cross-diamond-hexagonal search (CDHS) algorithm. It employs a smaller cross-shaped pattern before the first step of CDS and replacing the diamond-shaped pattern with hexagonal search patterns (HSP) in subsequent steps. So CDHS works better than CDS, DS, 4SS.

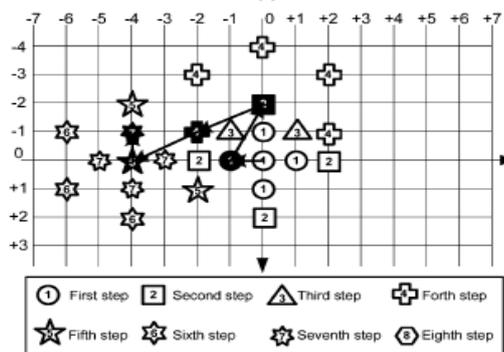


Figure . Cross Diamond-Hexagon Search

Jo Yew Tham [14] proposed a unrestricted center biased diamond search (UCBDS). This algorithm is more robust & efficient than the TSS, FS, 4SS. The UCBDS technique is mainly employs for fast suboptimal block based motion estimation.

Shiping Zhu [16] proposed a novel fast block-matching called cross-hexagon search algorithm (NHEXS). The new hexagon search consists of two patterns: cross-based and hexagon-based patterns. As the motion vectors distribution possesses cross center biased characteristics in the central 4 x 4 pixels area with two cross-shaped patterns: small-cross-shaped (SCSP) and large-cross-shaped (LCSP) are proposed as the first two initial steps to the hexagon-based search. Also there are two different sizes of hexagon search patterns: large and small hexagon patterns. The large hexagon pattern consists of not only the 7 check points in classical large hexagon pattern, but also the two edge points (up and down). Therefore, the new large hexagon pattern consists of 9 search points which realizes a distinct search speed gain without increasing computational complexity of large hexagon search algorithm.

Next some approaches on Quadtree and Weighted Finite Automata are discussed.

Guojun Lu[24] explained a method on quadtree partitioning, in which image is partitioned into four equally sized child range blocks. These four children are named according to their direction in the parent range block – Northwest (NW), Northeast (NE), Southwest (SW), and Southeast (SE). This process repeats, starting from the whole image and continuing until the range blocks are small enough to be matched.

Ullrich Hafner [25] used bintree to address pixels in an image. There is a trivial recursive algorithm to obtain the bintree representation of an image. First, divide the image recursively into halves until there is only one pixel left. We apply alternatively a horizontal and a vertical subdivision in order to get an unique binary partitioning of the image. Then, each pixel of a $2^d \times 2^d$ image can be addressed with a word w_1, w_2, \dots, w_{2d} of the alphabet summation of $2d$ denoting the position of the pixel in the corresponding bintree.

Zhuhan Jiang [26] proposed an extended WFA. An extended WFA is a preferred WFA to represent a target image because it has less number of states, thus requiring typically less space, which is a compression purpose, for the storage of the WFA. And used resolution wise mapping which states, a resolution-driven mapping is a mapping that maps a finite sequence of multiresolution images into another multiresolution image and are indistinguishable at the same resolution. A resolution-driven mapping is said to be a resolution-wise mapping if the resulting image at any given resolution is completely determined by the finite sequence of multiresolution images at the same resolution.

Karel culik [29] proposed his work which is an improvement over Culik-Kari's recursive inference algorithm which is a linear combination of available states. Here author used the transformations like rotation, flipping, negation of states as linear combination and named it Generalized automata, but had a drawback for storing more information for each edge. It has advantage of working well with all kind of images and created only one GWFA for a color image.

Ghim Hwee Onget [30] had proposed a binary approach. The idea is to first divide the current block/state horizontally into two (rectangular) substates and later divide again vertically to produce four (square) states. This effectively separates the previous quadtree partitioning into two steps: a horizontal cut followed by a vertical cut. The intention behind this separation is to better utilize the spatial correlations among image segments, by looking at 50% effective area. However, this involves some back-tracking.

F. Katritzkea [35] explained advanced technique of partitioning: light HV partitioning since it is a special case of the HV partitioning method. This method requires the efficient storage of additional data. But the improvement in image quality must be high enough to exceed the extra cost.

VII. PERFORMANCE MEASURES

The performance of fractal video compression depends mainly on four parameters. These parameters are the encoding speed, computational complexity, Signal to Noise Ratio (*PSNR*) (representing the quantitative visual quality of the video sequence), and finally the bit-rate. Reducing the computational complexity is the primary hurdle we need to increase the encoding speed.

A. Mean Square Error-

The MSE(Mean Square Error) is computed for each displacement position (i,j) within a specified search range in the reference image and the displacement that gives the minimum value of MSE is the displacement vector which is more commonly known as motion vector[38].

Let I and K be two monochrome images with size m x n[17]. MSE can be given as-

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

B. Mean of Absolute Difference-

Like the MSE criterion, the mean of absolute difference (MAD) also makes the error values as positive, but instead of summing up the squared differences, the absolute differences are summed up[38]. The MAD measure at displacement (i,j) is defined as-

$$MAD = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]$$

The MAD criterion requires computations of m x n subtractions with absolute values and m x n additions for each candidate block at each search position. The non-presence of multiplications makes this criterion computationally more attractive and facilitates easier hardware implementation[38].

C. Peak Signal to Noise Ratio-

The PSNR is the most commonly used as a quality measure of reconstruction of lossy compression codes. It measures the amount of noise in the regenerated video. It is observed that maximum PSNR is achieved at less size of range blocks. Because with higher sized range blocks, more image area is covered and hence during encoding process, error is introduced in coefficient determination. This error affects the regenerated image and thus resultant image is merely an approximate representation of the original image and so original video. But with less sized range blocks, lesser area of image is covered and size of domain blocks is also less and hence error is minimized in coefficient determination procedure[17].

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

MAX is the maximum pixel value of the image. When pixels are represented with *b* bits per sample, MAX is $2^b - 1$.

D. Compression Ratio-

It is observed that for least sized range blocks, compression ratio is lesser and as range size is increased, compression ratio also increases. This behavior is observed for larger range sized blocks, the amount of details in reconstructed video is lesser than those with lesser range sizes[17]. The generalized formula for compression ratio is given as-

$$\text{Compression ratio} = \frac{\text{Size of uncompressed image}}{\text{Size of compressed image}}$$

VIII. KEY ISSUES IN FRACTAL VIDEO COMPRESSION

The advantages of fractal compression attracts researchers more as it is new technology, While challenges to implement compression algorithm is also an issue. Fractal compressions has many key issues. Some of them are as follows-

A. Encoding Time -

This is the most challenging issue in fractal compression. Fractal compression gives high compression ratio. But encoding time is very high due to finding best matched block and no. of search points.

B. Partitioning Scheme-

Partitioning scheme can be Quad-tree partitioning or Binary partitioning. Which partitioning scheme can be best in order to completely partition an image to get enough quality.

C. Domain Pool Minimization-

The major drawback is that fractal encoding is complex and time consuming to search for the best-matching block in a large pool of domain blocks. As dividing the entire image into domain blocks, stored as domain pool, becomes a problem. Because domain pool blocks are overlapping blocks. If there is large no. of domain blocks then required space will be more and this seriously embarrasses the fractal image coding method's application into the practice[42].

D. Compression Ratio-

Compression ratio is directly proportional to Size of Range blocks. For less size of range blocks, so compression ratio is lesser and as range size is increased, compression ratio also increases.

E. Complexity of Algorithm-

In fractal compression, the algorithm to implement the compression, becomes more complex. As it requires to reduce encoding time as well as no. of search point to work in search area. To implement such algorithm is a complex task.

IX. ADVANTAGES AND DISADVANTAGES OF FRACTAL COMPRESSION

It is a promising technology, though still relatively immature. In common implementations, it is a block-based, lossy compression method. Fractal image interpolation may prove useful in multimedia applications[47]. Some of the main advantages and disadvantages are as given below-

A. Advantages of Fractal Compression

- Good mathematical encoding frame.
- Resolution independence.
- High compression ratio.
- Decompression is fast.

B. Disadvantage of Fractal Compression

- Computationally Intensive
- Complex algorithm

REFERENCES

- [1]. M.Barnsley and H.Rising, "Fractals Everywhere", Academic Press, San Diego,CA,USA, 1993
- [2]. A.E.Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations" IEEE 1992
- [3]. Shiping Zhu,Liang Yu,Kamel Bellouata, "An improved fractal image coding algorithm based on adaptive threshold for quadtree partition" The International Society for Optical Engineering, 2008
- [4]. Hai Wang, "Fast Image Fractal Compression with Graph-Based Image Segmentation Algorithm" 2010
- [5]. 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, 1981.
- [6]. R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technology vol. 4, 1994
- [7]. Donglai Xu, Chris Bailey and Reza Sotudeh, "An Improved Three-Step Search Block-Matching Algorithm for Low Bit-Rate Video Coding Applications", 1998 IEEE
- [8]. L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technology vol. 6, 1996
- [9]. Ming-gang Liu, Chao-huan Hou, "A Fast Block-Matching Motion Estimation Algorithm Based on Spatial-Temporal Motion Vector Correlation" International Symposium on intelligent Multimedia, Video and Speech Processing 2001

- [10]. Chun-Ho Cheung and Lai-Man Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, December 2002.
- [11]. Queen Mary, "A Novel Hexagonal Search Algorithm for Fast Block Matching Motion Estimation", EURASIP Journal on Applied Signal Processing 2002
- [12]. Kamel Belloulataa, Shiping Zhub, Jun Tianb, Xiaodong Shenb, " A Novel Cross-Hexagon Search Algorithm for Fast Block Motion Estimation " 2011 IEEE
- [13]. Chun-Ho Cheung and Lai-Man Po, " Novel Cross-Diamond-Hexagonal Search Algorithms for Fast Block Motion Estimation", IEEE Transactions on Multimedia, Vol. 7, No. 1, February 2005
- [14]. Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath, and Ashraf Ali 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, August 1998
- [15]. Shiping Zhu, Yangshuan Hou, Zaikuo Wang, Kamel Belloulata, A Novel Fractal Video Coding Algorithm Using Fast Block-matching Motion Estimation Technology" 2010 IEEE
- [16]. Meiqing Wang, Rong Liu, Choi-Hong Lai, Adaptive Partition and Hybrid Method in Fractal Video Compression, ELSEVIER 2006.
- [17]. Kulkarni M.V. And Kulkarni D.B., "Analysis of Fractal Video Coding Using Fixed Partitioning Scheme" Journal of Signal and Image Processing - Volume 3, Issue 4, 2012.
- [18]. Mr.M.Manikandan, Mr.P.Vijayakumar, Mr.N.Ramadass, Motion Estimation Method for Video Compression –An Overview, IEEE 2006
- [19]. M. S. Lazar and L. T. Bruton, "Fractal Block Coding of Digital Video"
- [20]. Chris Urmson and Ken Ferens, "Video Compression Through Fractal Block Coding", 1998 IEEE
- [21]. Y.G. Wu, G.-F. Huang, "Motion vector generation for video coding by gray prediction", IET Comput. Vis., 2011, Vol. 5, Iss. 1, pp. 14–22
- [22]. William Navas, Hamed Parsiani, "Recent Advances in Weighted Finite Automata Fractals", 1999 IEEE
- [23]. David J. Jackson, Wagdy Mahmoud, William A. Stapleton, Fast Fractal Image Compression using Quadtree Recomposition, ELSEVIER 1995.
- [24]. Guojun Lu "Fractal-Based Image and Video Compression" CRC Press LLC, 2001
- [25]. Ullrich Hafner, Stefan Frank, Michael Unger, Jurgen Albert, " Hybrid Weighted Finite Automata for Image and Video Compression", March 1997
- [26]. Zhuhan Jian, Bruce Litow, Olivier de Vel, "An Inference Implementation Based on Extended Weighted Finite Automata" 2001 IEEE
- [27]. Culik K and Kari J, "Inference algorithm for WFA and image compression", Fractal Image Compression: Theory and Application, edr. Fisher Y, Springer, New York 1995
- [28]. Hafner U, "Refining image compression with weighted finite automata", Proceedings of Data Compression Conference, eds. Storer J and Cohn M, (1996)359-368.
- [29]. Karel Culik, Peter C. von Rosenberg, "Generalized Weighted Finite Automata Based Image Compression", vol.5, no.4, Springer Pub. Co. 1999
- [30]. Jiirgen Albert, Stefan Frank, Ullrich Hafner*, Michael Unger, "Video Compression with Weighted Finite Automata",1997 IEEE
- [31]. Ghim Hwee Ong And Kai Yang, "A Binary Partitioning Approach to Image Compression using Weighted Finite Automata for Large Images" 2006 Elsevier
- [32]. Marian Mindek, "Finite State Automata and Image Recognition", Technical University of Ostrava, 2004
- [33]. Helmut Jürgensena,b, Ludwig Staigerc, Hideki Yamasakid, "Finite automata encoding geometric figures", 2007 Elsevier
- [34]. Ullrich Hafner, "Image and Video Coding with Weighted Finite Automata"
- [35]. F. Katritzkea, W. Merzenich, M. Thomas, " Enhancements of partitioning techniques for image compression using weighted finite automata ", 2003 Elsevier
- [36]. Zhuhan Jianga, Olivier de Velb , Bruce Litowc, "Unification and extension of weighted finite automata applicable to image compression", 2002 Elsevier
- [37]. N.A.Koli and M.S.Ali, "A Survey on Fractal Image Compression Key Issues", Information Technology Journal 2008.
- [38]. Video Coding And Motion Estimation- Block Based Motion Estimation Algorithms, Version 2 ECE IIT, Kharagpu Dr. K. Kuppusamy, R.Ilackiya, "Fractal Image Compression & Algorithmic Techniques", 2013
- [39]. Meenu Bansal, Sukhjeet K. Ranade, "A review on fractal image compression", 2012
- [40]. Miroslav Galabov, " Fractal Image Compression", CompSysTech - 2003
- [41]. Kamel Belloulataa, Shiping Zhub and Zaikuo Wangb, "A Fast Fractal Video Coding Algorithm Using Cross-Hexagon Search for Block Motion Estimation"
- [42]. Y. Fisher, "Fractal encoding with quadtrees," in Fractal Image Compression: Theory and Applications to Digital Images, Springer-Verlag, New York, USA, 1995
- [43]. T. Ochotta and D. Saupe, " Edge-based partition coding for fractal image compression", 2004
- [44]. K. Belloulata, "Fast fractal coding of subbands using a non-iterative block clustering", 2005
- [45]. Information Technology, Coding of Audio Visual Objects - Part 2: Visual, ISO/IEC, (MPEG-4 Visual), 1999
- [46]. John Kominek, " Advances in fractal compression for multimedia applications", Multimedia Systems, Springer-Verlag 1997