Modified Three-Step Search Block Matching Motion Estimation and Weighted Finite Automata based Fractal Video Compression

Shailesh D. Kamble[1], Nileshsingh V. Thakur[2], and Preeti R. Bajaj[3]

[1]*Computer Science & Engineering, Yeshwantrao Chavan College of Engineering, India*
[2]*Computer Science & Engineering, Prof Ram Meghe College of Engineering & Management, India*
[3]*Electronics Engineering, G. H. Raisoni College of Engineering, India*

***Abstract* — The major challenge with fractal image/video coding technique is that, it requires more encoding time. Therefore, how to reduce the encoding time is the research component remains in the fractal coding. Block matching motion estimation algorithms are used, to reduce the computations performed in the process of encoding. The objective of the proposed work is to develop an approach for video coding using modified three step search (MTSS) block matching algorithm and weighted finite automata (WFA) coding with a specific focus on reducing the encoding time. The MTSS block matching algorithm are used for computing motion vectors between the two frames i.e. displacement of pixels and WFA is used for the coding as it behaves like the Fractal Coding (FC). WFA represents an image (frame or motion compensated prediction error) based on the idea of fractal that the image has self-similarity in itself. 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. The proposed MTSS block matching algorithm is based on the combination of rectangular and hexagonal search pattern and compared with the existing New Three-Step Search (NTSS), Three-Step Search (TSS), and Efficient Three-Step Search (ETSS) block matching estimation algorithm. The performance of the proposed MTSS block matching algorithm is evaluated on the basis of performance evaluation parameters i.e. mean absolute difference (MAD) and average search points required per frame. Mean of absolute difference (MAD) distortion function is used as the block distortion measure (BDM). Finally, developed approaches namely, MTSS and WFA, MTSS and FC, and Plane FC (applied on every frame) are compared with each other. The experimentations are carried out on the standard uncompressed video databases, namely, akiyo, bus, mobile, suzie, traffic, football, soccer, ice etc. Developed approaches are compared on the basis of performance evaluation parameters, namely, encoding time, decoding time, compression ratio and Peak Signal to Noise Ratio (PSNR). The video compression using MTSS and WFA coding performs better than MTSS and fractal coding, and frame by frame fractal coding in terms of achieving reduced encoding time and better quality of video.**

***Keywords* — Fractal Coding, Motion Wstimation, Three Step Search, Cross Hexagon Search, Encoding Time, Compression Ratio.**

I. Introduction

Video compression techniques deal with the lossy or lossless data compression for the series of image sequences. Gray and color image sequences are the customers for the video compression techniques. From the preprocessing point of view different color spaces [1] can be used for the image sequences. With the recent development in the area of internet and multimedia technology with moving video, a color image sequences processing plays an important role. Each color plane is represented by 8 bits/pixel in a RGB color space therefore a RGB color image is represented by 24 bits/ pixel. The intra-frame and inter-frame coding is used to reduce the spatial and temporal data redundancy present in the image sequences.

Block matching motion estimation plays an important role in inter-frame coding technique to reduce temporal redundancy present in the series of image sequences. Therefore it is a most popular and efficient technique for computing the motion vectors which has been used by various coding standards. Video coding standards are related to the organization, such as ITU-T Rec. H.261, H.263, ISO/IEC MPEG-1, MPEG-2, MPEG-4, and recent progress is H.264/AVC. Different block searching approaches are available for motion estimation [2-7] between the successive frames. A block-matching algorithm [5-6] can be used to improve the quality of video and performance of coding process [7].

Block-matching algorithms are the most successful approaches for motion estimation in the video compression technology because of it easy to understand and with some efforts it can be implemented easily. The simplest full search (FS) block matching algorithm provides a full exhaustive search within the search window for searching the optimal solution. Therefore to reduce the computational cost of FS, many block matching motion estimation algorithms are proposed such as three step search [8], 2D logarithmic search [9], orthogonal search [10], cross search [11], binary search [12], new three-step search [13], the four-step search [14], the block-based gradient descent search [15], the diamond search [16], the cross-diamond search [17], efficient three step search [18], etc. All these search algorithms employs a rectangular and diamond search patterns having the center-biased motion vector distribution characteristics [19-20]. Hexagon-based search algorithm which employs a two hexagon-based pattern i.e. large and small, and results in fewer search points is proposed [21]. Novel Cross-Hexagon based Search algorithm [6] consists of two cross shape patterns and two hexagon-based patterns. Modified partial distortion criterion (MPDC) used a certain block of pixels which improve the computations [22]. New Cross-hexagon search (NHEXS) algorithm [23-24] consist of two cross search patterns and hexagon search patterns which is similar to fast block-matching Motion Estimation [6].

Block matching Motion estimation is widely used in the video application areas such that rain pixel recovery for videos, video compression, medical video processing [25-26], object tracking and

DOI: 10.9781/ijimai.2017.444

surveillance [27], etc. Motion estimation means displacement of pixels position from one frame to another frame which gives the best motion vector (MV) [28]. In Block matching algorithm, each frame is divided into a fixed sized macro blocks and each corresponding macro blocks in the current frame are then compared with the adjacent neighbor macro blocks in the previous frame to estimate a motion vector (MV). The estimated MV specifies the displacement of a macro block from one position to another position in a previous frame as shown in Fig. 1. Motion vector for the block Bf(x, y) is computed as (+1,-1) i.e. MV-Bf(x,y) = (+1,-1). Still there is a scope of improvement in this field for implementing the fast block matching motion estimation algorithm.

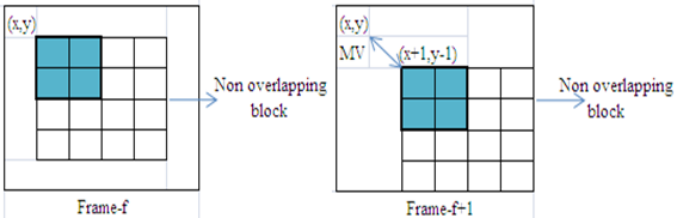


Fig.1. Block matching motion vector estimation.

Using Weighted Finite Automata (WFA) the weights are assigned to state transitions proposed by [29-30], WFA provide a powerful tool for image representation as a WFA and compress the image in term of a good compression ratio. The inference algorithm for WFA subdivides an image into a set of non-overlapping range blocks and then separately approximates each range block with a linear combination of the domain block. WFA coding techniques is based on the idea of fractal coding that an image has self-similarity in itself [31-34] i.e. WFA coding is similar to the fractal coding approach.

This paper discusses a modified three step search algorithm for block matching motion estimation and WFA coding approach for color video compression. The paper is organized as follows. Section 2 reviews Three Step Search algorithm. New cross hexagonal search algorithm discussed in section 3. The proposed MTSS presented in section 4 and explains the video compression process using MTSS and WFA coding. Section 5 presents experimental results of proposed MTSS approach in comparison with TSS, ETSS and NTSS block matching algorithm and further the results of proposed MTSS and WFA coding is compared with MTSS and fractal coding. Finally, Section 6 presents the conclusion and future scope based on proposed work.

II. Three Step Search Algorithm

Three step search (TSS) proposed by Koga et al. [8] is one of the earliest attempt to implement a fast motion estimation algorithm. The computational cost of this algorithm is less in terms of MAD matching criterion, average search point and computation required as compared to the full search algorithm. In this algorithm first step is to define a fixed size of 16 ×16 macro block, searching parameter p pixels in all four directions and 9×9 search window in the central part of the 16 ×16 macro block for searching the best match. The TSS algorithm is summarized as follow:

Step 1: Plot 9 points in the search window at the equal distance of step size s=4 and checked 9 points on the 9×9 search window. .

Step 2: Step size is divided by 2 i.e. s=2 and check 8 points to generate 5×5 square shape pattern. If minimum block distortion measure (BDM) is one of the nine points of generated search window then this point consider as a center point in the step 3.

Step 3: Step size is divided by 2 i.e. s=1 and check 8 points to generate 3×3 square shape pattern and search will be terminated.

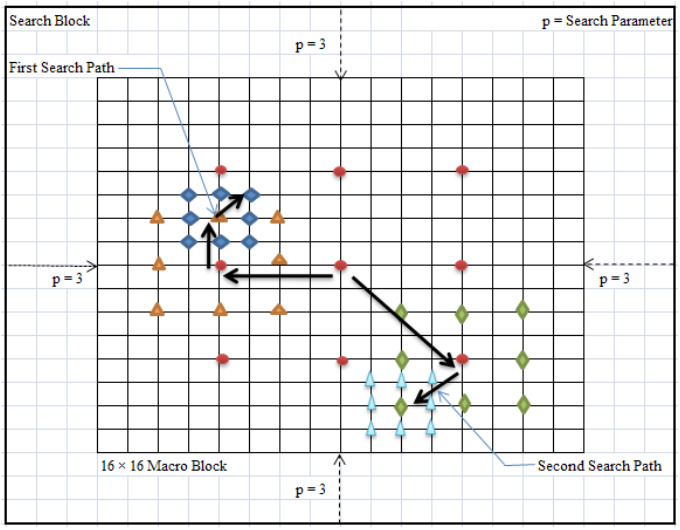


Fig.2. Two different search paths for three step search motion estimation.

The minimum BDM point found at the 3×3 square shape pattern is the final motion vector. Fig. 2 shows the two different search path of TSS for estimating a motion vector within search window. TSS can be easily extended up to n-step search for larger search window. The number of checking points required for TSS is 25.

III. New Cross Hexagonal Search Algorithm

Kamel Belloulata et al. [24] proposed a novel fast block matching algorithm called new cross hexagon (NCHEXS) pattern based search using two small/large cross shape(SCSP/LCSP) pattern as a first three initial steps and two small/large cross hexagon shape pattern (SHSP/LHSP) as a subsequent steps of search. In the beginning of this algorithm, initialize SCSP by plotting 5 points at the center of 16×16 macro blocks [28]. The algorithm is summarized as follow:

Step 1: If minimum BDM point found at the center of the small cross shape pattern then stop searching otherwise go to step 2.

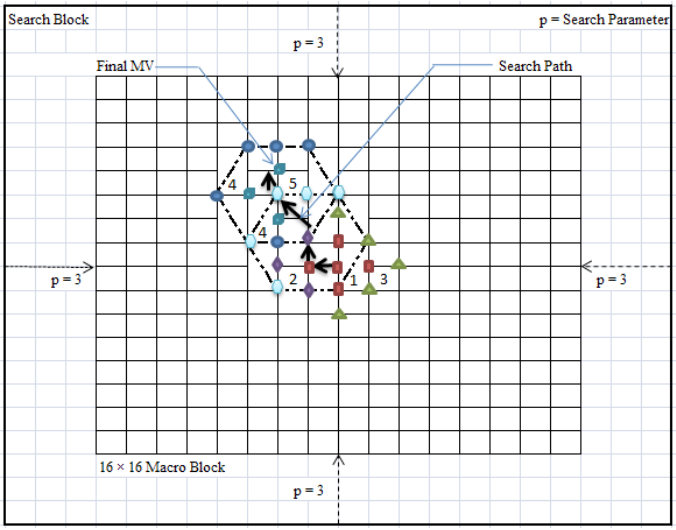
Step 2: If minimum BDM point found at the center of newly formed small shape pattern then stop searching otherwise go to step 3.

Step 3: Check the another 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.

Step 4: A new large hexagonal shape pattern is formed by considering a center point as minimum BDM point found in previous step. If minimum BDM point found at the center of new large hexagonal pattern then go to step 5 otherwise again form a new large hexagon pattern i.e., repeat step 4.

Step 5: If minimum point found at the center of large hexagonal pattern then large hexagonal pattern shifted to small hexagonal pattern and find best motion vector in small hexagon shape pattern

Fig. 3 shows the search path of NCHEXS. The number of checking point required for NCHEXS is from 5 to 8 for best case which is better than all the techniques available for estimating a motion vector.



SCSP: 1st step: , *2nd step:* , *LCSP: 3rd step:* , *LHSP: 4th step:* & ,*SHSP: 5th step:* *Final MV.*

Fig.3. Search path in New Hexagon search motion estimation.

IV. Proposed Approach

The main objective of proposed work is to develop a mechanism for color video compression based fractal coding technique using Modified three-step search (MTSS) and weighted finite automata (WFA) coding. Therefore, fractal coding based color video compression using MTSS block matching algorithm and WFA coding is proposed.

In color images, Each R, G and B components contains 8 bit data and also every color image contains lots of redundancy. In this approach, each frame is converted into the YC_bC_r color space i.e. most suitable color space for video processing and the first component of the frame in YC_bC_r color space is treated as gray scale image. The first component of YC_bC_r i.e. luminance component is sensitive to human eye and similar to grey scale image and remaining two chrominance component consist of color information and human eye is not that much sensitive for these two planes. Therefore, the focus is on the compression of the first luminance plane. Each gray scale frame is divided into a fixed/equal sized macro blocks. The MTSS approach is used for estimating a motion vector in the current frame with respect to the previous frame. The predicted frame for the current frame is then created from the previous frame with assigned motion vectors. Encoding and decoding of the previous frame is performed using WFA or FC. Encoding and decoding of the difference between the predicted frame and the current frame is, also, performed using WFA or FC. The first frame is encoded and decoded using WFA/FC and then the predicted frame for the second frame is formed from the decoded first frame using MTSS. Then, the predicted frame is encoded using WFA/FC and decoded to form the predicted frame for the third frame. This process is repeated for all the frame sequences. The process flow of proposed MTSS and WFA/FC coding approach for video compression is shown in Fig. 4.

A. Modified Three Step Search (MTSS) Algorithm

The proposed MTSS approach is used to calculate the motion compensation prediction error (MCPE) between two consecutive frames. In MTSS, 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 is used in the center part of search window to exploit central biased characteristic of Motion Vector (MV) in video sequences. The proposed MTSS approach

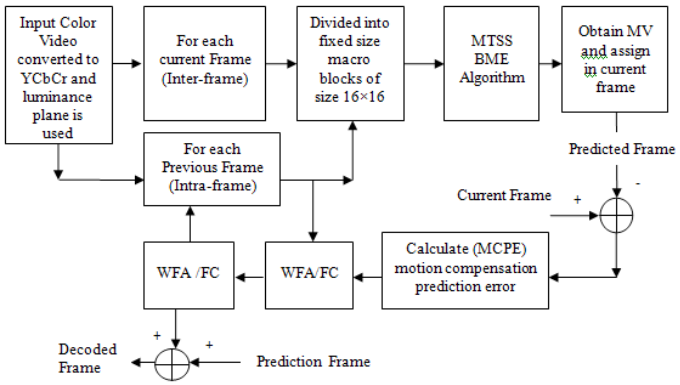


Fig.4. MTSS and WFA/FC video coding Process Flow.

consists of TSS and NCHEXS approach. TSS and NCHEXS employ square and hexagon based shape pattern of different sizes respectively. If minimum BDM point found at outer point of 9×9 search window then TSS will execute. Otherwise NCHEXS will execute at the central part of the search window. Fig. 5 shows the search pattern used in first step of MTSS proposed approach. The MTSS can be summarized as follows:

Step 1: Total 9+4 points are checked, if minimum BDM point found at the center of 9+4 points then stop searching otherwise go to step 2.

Step 2: If minimum BDM point found at the outer part of search window then search process is same as TSS discussed in section 2 otherwise go to step 3.

Step 3: If minimum BDM point found at the 4 outer points of small cross search pattern (SCSP) then search process is same as new cross hexagon search (NCHEXS) discussed in section 3.

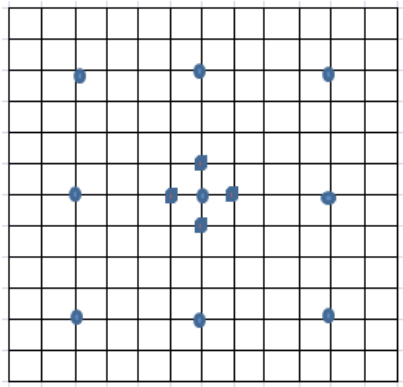


Fig.5. Search pattern used in first step of modified TSS (MTSS).

Block diagram and two search path of proposed MTSS approach are shown in Fig. 6 and 7 respectively. The numbers of checking points required for MTSS approach are 13 (9+4) points for best case i.e. stationary block is shown in Fig. 5, 13 to 29 (9+4+8+8) points for average case and more than 29 points for worst case is shown Fig. 7. While checking points are needed with the new three step search (NTSS) algorithm are 17 for best case, 17 to 33 points for average case and 33 checking points are required for worst case. In TSS, 25 checking points are needed in all the cases.

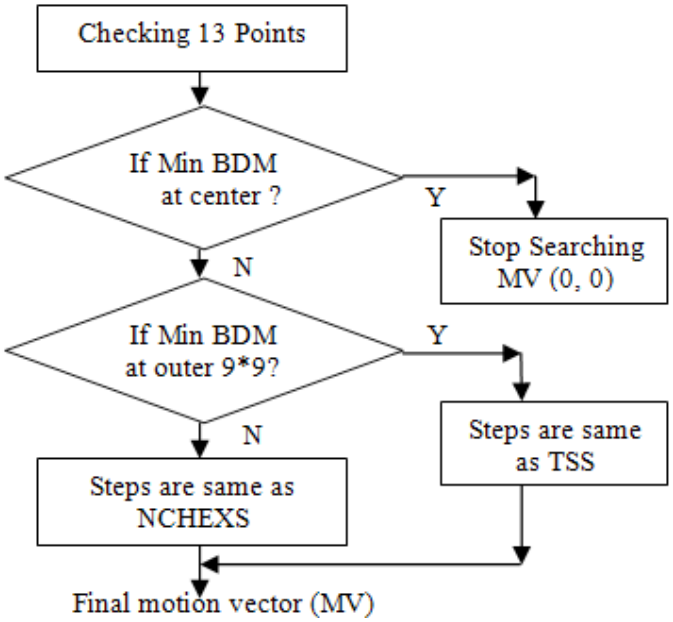


Fig..6. Block schematic of the Modified TSS (MTSS).

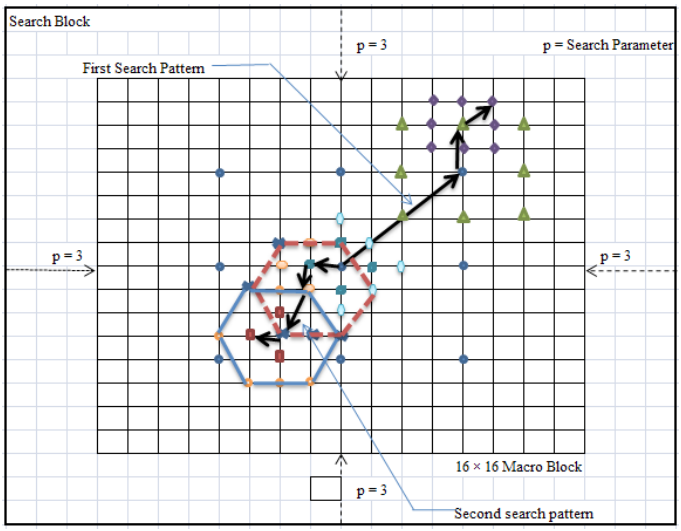


Fig.7. Two Search paths in modified TSS.

The concept of intra-frame and inter-frame coding are used for exploiting the self similarity present in the frame. The intra-frame coding is similar to the individual frame by frame coding i.e. individual frames are coded independently. The first frame in a video sequence is always an intra-frame because there is no previous data related to the first frame. It is not necessary in a video sequence always a first frame is an intra-frame. Intra-frame occurs anywhere in video sequences when the scene of video sequence is totally changed. On the other side in inter-frame coding, the previous frame and current frame is coded using proposed MTSS block matching motion estimation algorithm and WFA coding. The previous and current frames are extracted from the video sequence and apply the proposed MTSS algorithm on each fixed sized blocks. The obtained motion vectors (MV) are assigned to the current frame *i.e.,* the newly formed predicted frame. Then calculate the motion compensated predicted error (MCPE) frame by deleting the current frame from newly formed predicted frame. Finally apply the WFA encoding and decoding on them as shown in block schematic Fig. 4. The WFA decoded frame is considered as a previous frame for the new subsequent current frame and same process is repeated.

B. Proof-Outlines of the Parameters used in Modified Three-Step Search (MTSS) algorithm

In general, videos consist of different number of frames. Consecutive frames have the huge spatial redundancy. Occurrence of the change in next frame is very less in comparison with the previous frame. Most of the part of next frame is similar to the previous frame. Therefore, the change, in general, may occur within the area of 3×3 or 5×5 or 7×7 or 9×9 neighborhood of pixels. Hence, to measure the similarity between two consecutive frames, it is better to use the bigger size search window to grab most of the area of macro block size (i.e. 16×16). So, initially, 9×9 search window (centre-biased) is selected. Now, 3 or 4 pixels are left on every side of 9×9 search window. As unexplored area consist of 3 or 4 pixels, the scope of next search window of size (3×2+1)×(3×2+1) i.e. 7×7 or (2×2+1)×(2×2+1) i.e. 5×5 exists. Here, we have used 5×5 search window (centre-biased). Now, unexplored area consist of 1 pixel, the scope of next search window of size (1×2+1)×(1×2+1) i.e. 3×3 exists. Hence, we have used 3×3 search window (centre-biased).

C. Weighted Finite Automata Representation

A finite automaton is a mathematical model used to represent an image. It exploits the self-similarity, i.e. redundancy, in an image. WFA constitute an extended version of finite automata, with transitions labeled as the addresses of the sub-images and weights assigned from one state to another state. WFA are used to represent images in a 6-tuple form, i.e. M= (Q, ∑, W, I, F, q_0), where,

Q is a finite set of states, i.e. {q_0, q_1, q_2 ...q_n};

∑ is a finite set of input symbols/quadrant addresses, i.e.

∑= {0, 1, 2, 3} for quadtree WFA,

∑= {0, 1} for bintree WFA, and

∑= {0, 1, 2, 3, 4, 5, 6, 7, 8} for nona-tree WFA;

W is the weight function between two states, i.e.

W Є Q×Q → R for W (q_0, q_1) = R, where q_0, q_1 Є Q and R is a real number, i.e. the weight between states q_0 and q_1;

I is the initial configuration of states Q → R and indicates which states correspond to the entire image; I (q_0) = 1 and I(q_i) = 0, $q_0 \neq q_i$, where q_0, q_i Є Q and I =1, 2, 3....n;

F is the final configuration of states Q → R, e.g. F (q_0) = f (Є), where q_0 Є Q and f (Є) is the average intensity (greyness) of the entire image;

q_0 is the initial state of the WFA, i.e. the entire original image: q_0 Є Q

The transition from (q_0, 1) = q_1 Є Q×∑ →Q if W (q_0, q_1) = R or W (q_0, q_1) ≠ 0 implies that there is a transition from q_0 to q_1 on the input symbol labelled by 1. Further, W_i(q_0, q_1) denotes a weighted transition from q_0 to q_1 on the input symbol, i.e. state or sub-image i.

The basic principles of the WFA approach are somewhat similar to those of the fractal image compression approach based on PIFS (partitioned iterated function systems). Both the approaches use the fact that images used in practice have a certain amount of self-similarity present in images to achieve compression. In other words, a sub-image of the image to be compressed may be similar to another sub-image of the same image, except perhaps for size, contrast, or brightness. The main difference between the two approaches is that the PIFS-based fractal compression uses affine transformations to find the self-similarity of sub-images, while on the other hand, WFA finds a sub-image as a weighted linear combination of other states/sub-images. The real value (weight) and quadrant address assigned with each transition along an edge in the WFA indicate how each state in the WFA is expressed as a linear combination of the other states:

$$(s_i)q = Wq(s_i, s_0)s_0 + Wq(s_i, s_1)s_1 + \cdots + Wq(s_i, s_n)s_n \quad (1)$$

Where s_i is the image associated with state s_i and $(s_i)q$ indicates the address of quadrant q of state s_i .

WFA provide a powerful tool for image generation and compression. First, the image is subdivided into non-overlapping sub-images through a quadtree partitioning scheme. These sub-images are identical to those range blocks used in the PIFS-based fractal image compression approach. Next, one or more sub-images that are very similar or identical to the original image or to each range block/sub-image to be encoded separately are obtained from a domain block/sub-images present in domain pool, and a transition graph is constructed to describe the relationship between these sub-images and finally, the image to be encoded using WFA approach. The domain pool may contain all states or sub-images which could be generated from the given partitioning scheme. In general, the WFA uses an inference algorithm to construct a transition graph that is very similar to graphs used to represent finite state automata. The various states / sub-images of the finite state automata are then compressed to become the compressed image.

The process of decoding an encoded image with suitable example is discussed below. The image to be encoded at resolution 4×4 i.e. $2^{k=2} \times 2^{k=2}$ is given in Fig. 8 (a). The generated quadtree, WFA representation and its transition diagram representation are given in Fig. 8 (b), 8(c) & 8(d).

The WFA representations in the form of matrix are as follows.

$$Q=\{ S1 , S2 \} , \sum = \{ 0,1,2,3 \} , I = [1\ 0] , F = \begin{bmatrix} 2.2 \\ 2 \end{bmatrix},$$

$$W0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, W1 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, W2 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \text{ and } W3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

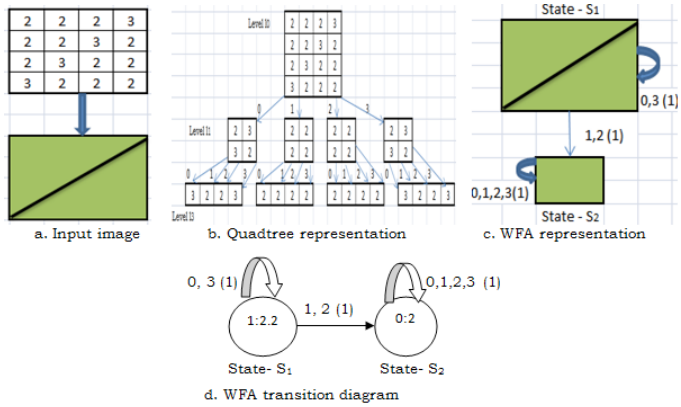


Fig.8. Process of WFA encoding/decoding.

WFA Encoding Process

The WFA encoding algorithm takes as input a grayscale image of size $2^k \times 2^k$ and gives as output the WFA representation of an image of size $2^k \times 2^k$ i.e. the initial configuration I, the final configuration F and weights W. In WFA all the four sub-image/ quadrants of an image are processed by approximating the quadrant/sub-image with the linear combination of all the existing states. If the quadrant of state/image is not a linear combination of the existing states then the quadrant is chosen as a new state and placed in the unprocessed state list; if not, then add transition and stores coefficients in the list. All the four quadrants/sub-images of a state/image are processed before moving to the next unprocessed state present in the list. Once all the new unprocessed states present in the unprocessed list have been processed, the algorithm terminates. The recursive WFA encoding algorithm is shown below.

Input : Image M of size $2^k \times 2^k$

Output: Generated WFA M representing the image M of size $2^k \times 2^k$

Step-1: n =0, Number of states in WFA

$W_q (s_i, s_j)$ =0, indicate the weight associated with the transition from state s_i to s_j with real value W_q.

$(s_i)_q$ = quadrant indicating address q of state s_i.

i=1, The index of first unprocessed state.

$I(s_0)$=1,Initial configuration of entire image

$F(s_0) = f(\epsilon)$ is the average grayness or brightness of entire image

Step-2: For every q $\in \sum$ = {0, 1, 2, 3} do

Sub-image I = $(s_i)_q$

Approximate the sub-image/quadrant with linear combination of the existing states/ images

Approximate I= $W_q(s_i, s_0)s_0 + W_q(s_i, s_1)s_1 + \ldots + W_q (s_i, s_n) s_n$

add transition and store coefficient in the list

else n = n + 1

Create new state s_n = I

Add transition from state s_i to new state s_n with labeled q and weight 1

set $F(s_n)$ = and $I(s_n)$ = 0

Append new state s_n to WFA

Step-3: i= i+1

if i <= n then goto Step 2

WFA Decoding Process

The decoding algorithm takes as input an encoded WFA represented by $n + n_0$ states, F and W created during encoding process as well as the resolution level k and returns as output decoded image of size $2^k \times 2^k$. An WFA decoding approach is discussed below.

Input: An encoded EWFA represented by $n + n_0$ states, F, W and integer k for an image resolution $2^k \times 2^k$

Output: Decoded image M of size $2^k \times 2^k$

n_0 =1, Number of initial states in WFA

n = Number of non initial state created during execution of encoding algorith in WFA

l_n = Level of quadtree generated from top l_0 to bottom l_n

$f(s_i, \epsilon) = F(s_i)$ return the value of average grayness or brightness of image associated with state s_i

$f(s_i, aw)$ is the function returns the value of state s_i at quadrant a=0,1,2,3 and length of input string generated by $\sum^*$

To decode each state in WFA call routine built_decoder (WFA M, int n)

```
{
f(s_i, ϵ)= F(s_i) where i= 1,2,...n
for (i = 1 to l_n – 1)
{
  for ( s = 1 to n + n_0)
  {
    for every input symbol or string w ∈ Σ^(i-1) do
    {
      for every quadrant in a image a ∈ Σ do
      {
        f(s_i, aw) =
      }
    }
  }
  Rearrange all values f(s_i, aw) calculated in matrix form
}
```

V. Experimental Results and Discussion

The proposed approach is implemented using MATLAB® 2013a (8.1.0.604). The experiments were carried out on a system with an Intel® Core™ i3 CPU (2.40 GHz). The experiments were carried out on color videos (i.e. Soccer, Suzie, Bus, Football, Xylophone, Paris, Traffic, Akiyo, Ice, and Mobile sequences) obtained from online resources i.e., http://media.xiph.org/video/derf/ (see Fig. 9). Standard input streams with different frame rates, lengths of sequences, and frame widths/heights were considered to demonstrate the performance of the proposed approach (see Table I).

Fig.9. Uncompressed input video sequences.

TABLE I

Specification of standard databases

Sequences	Frame Rate frame/s	Length of video (s)	Frame Width × Height
Soccer	20	6	176 × 144
Suzie	15	6	176 × 144
Bus	20	3	176 × 144
Football	15	6	176 × 144
Xylophone	30	5	640 × 480
Paris	29	13	352 × 288
Traffic	15	8	160 × 120
Akiyo	29	10	325 × 288
Ice	15	6	176 × 144
Mobile	29	10	352 × 188

A. Quality Measures

Inter-frame and intra-frame coding is used to eliminate the large amount of temporal and spatial redundancy exists in the video sequences and therefore, helps 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 2 ,3 and 4 respectively.

$$\mathrm{MAD}(i,j) = \frac{1}{N\times N}\sum_{i=0}^{i=N-1}\sum_{j=0}^{j=N-1}|Cij - Rij| \tag{2}$$

$$\mathrm{MSE}(i,j) = \frac{1}{N\times N}\sum_{i=0}^{i=N-1}\sum_{j=0}^{j=N-1}(Cij - Rij)^2 \tag{3}$$

$$\mathrm{MSE}(i,j) = \frac{1}{N\times N}\sum_{i=0}^{i=N-1}\sum_{j=0}^{j=N-1}(Cij - Rij)^2 \tag{4}$$

Where, N×N is the row and column of the macro block, C_{ij} and R_{ij} are the pixels value compared in 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 the macro block size results in more motion vectors and more macro blocks per frame. Therefore, improves a quality of motion compensated prediction error (MCPE). Most video coding standards used a macro block of size 16×16 and 8×8. The best/single motion vector is computed for each macro block in reference frame. On the other hand, total number of search point to find motion vector per frame is one of the key parameter in block matching motion estimation algorithm. The performance of video coding is measured in terms of compression ratio (CR), quality of video i.e. PSNR, encoding time and decoding time. The compression ratio (CR) is given by equation 5.

$$\mathrm{CR} = \frac{\text{Size of original video}}{\text{Size of compressed file}} \tag{5}$$

Therefore, compression ratio in percentage is computed from 5 and given by equation 6.

$$\mathrm{CR}\,\% = \big(1 - (1/\mathrm{CR})\big)\times 100 \tag{6}$$

When measuring the quality of compressed video, the peak signal-to-noise ratio is used. Sometimes mean squared error (mse) is also used, is given by 7.

$$\mathrm{mse} = \frac{1}{n}\sum(X_i - Y_i) \tag{7}$$

From this, the PSNR for an 8-bit grayscale image is defined by equation (8), where 255 is the maximum value for 8-bit pixel can assume.

$$\mathrm{PSNR(dB)} = 10\log_{10}(255^2/\mathrm{mse}) \tag{8}$$

B. Stepwise Results Obtained for Color Space Conversion and the Quad-tree Partitioning Scheme

The intermediate results based on color space conversion and Quadtree partitioning of first ten frames of "soccer" video are shown in Fig. 10. The input video "Soccer" was obtained from an uncompressed video database i.e., http://media.xiph.org/video/derf/. The first 10 frames are shown in Fig. 10(a). The YC_bC_r color space i.e., Y- luminance component sequences of the first 10 frames are shown in Fig. 10(b). The Y- luminance component sequences of the first 10 frames are then converted into gray frames, as shown in Figures 10(c). Fig. 10(d) shows first 10 frames of quadtree partitioning of gray frames. Note that in experimentation the standard input frame size must be converted into square of size $2^n \times 2^n$. This conversion involves shrinking and replicating the pixels for finding out the frame of size $2^n \times 2^n$. The presented approach initially converts the RGB sequence into the YC_bC_r color space sequence and finally convert Y- luminance component to grey scale for applying the quadtree partitioning.

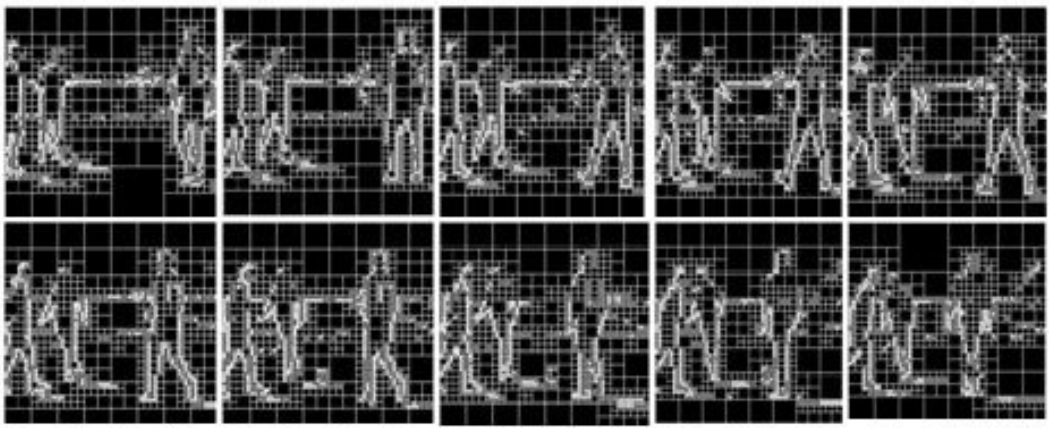
C. Evaluation and Comparison of Modified Three Step Search (MTSS) Block Matching Estimation

The mean absolute difference (MAD) minimum cost function is used as the Block distortion measure (BDM) given by equation 2. The performance of the MTSS block matching motion estimation algorithm is compared with the existing block matching algorithms i.e., TSS, NTSS and ETSS. Performance evaluation parameters used for comparison are- MAD and total number of search points/ check points required per frame and the results are shown in Fig. 11 and 12 respectively.

a. First ten color frames of soccer video.

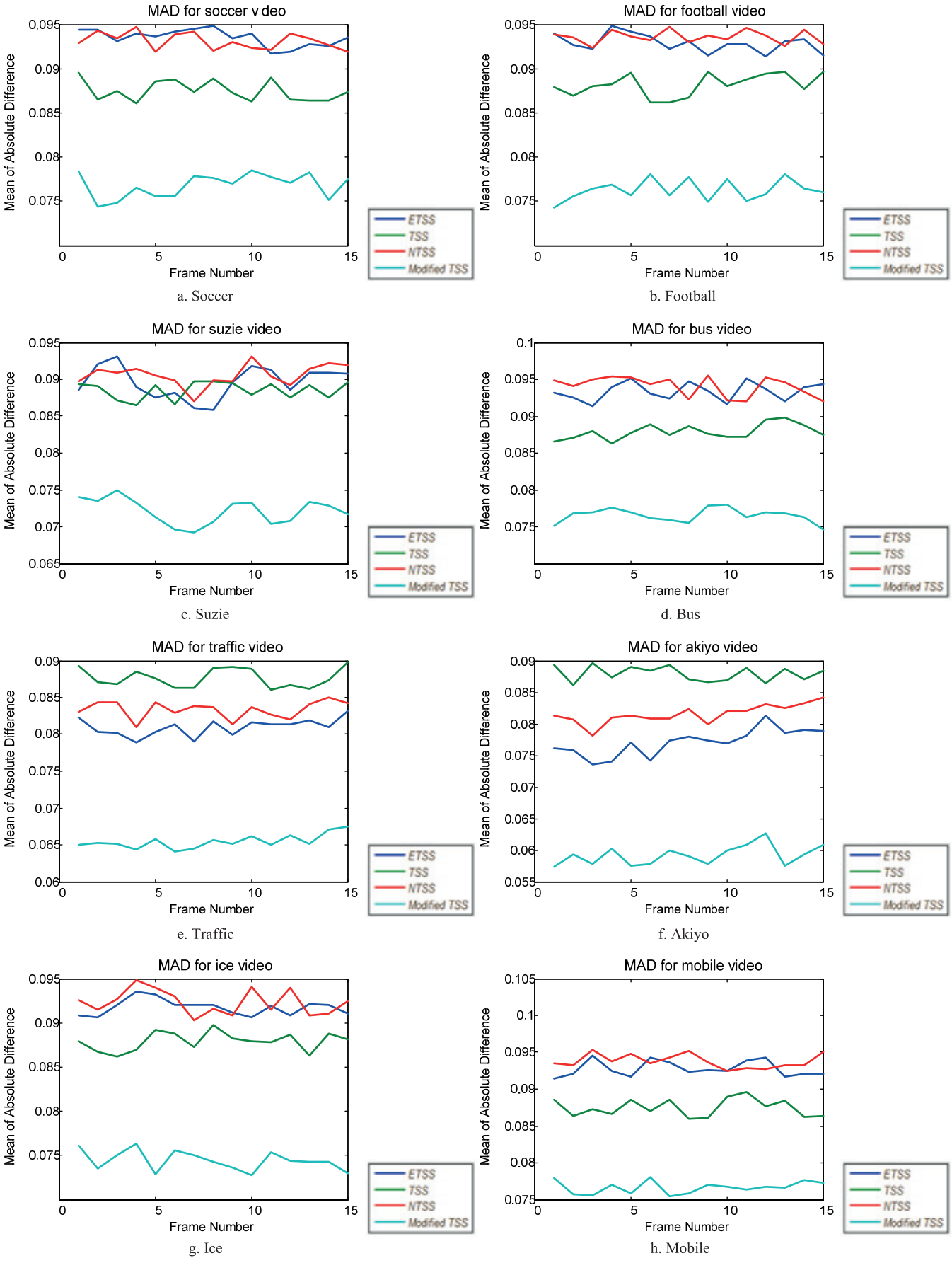
b. First ten YCbCr frames of soccer video.

c. First ten gray frames of soccer video.



d. First ten frames of quadtree partitioning

Fig.10. Intermediate results for the first ten frames of "soccer video".



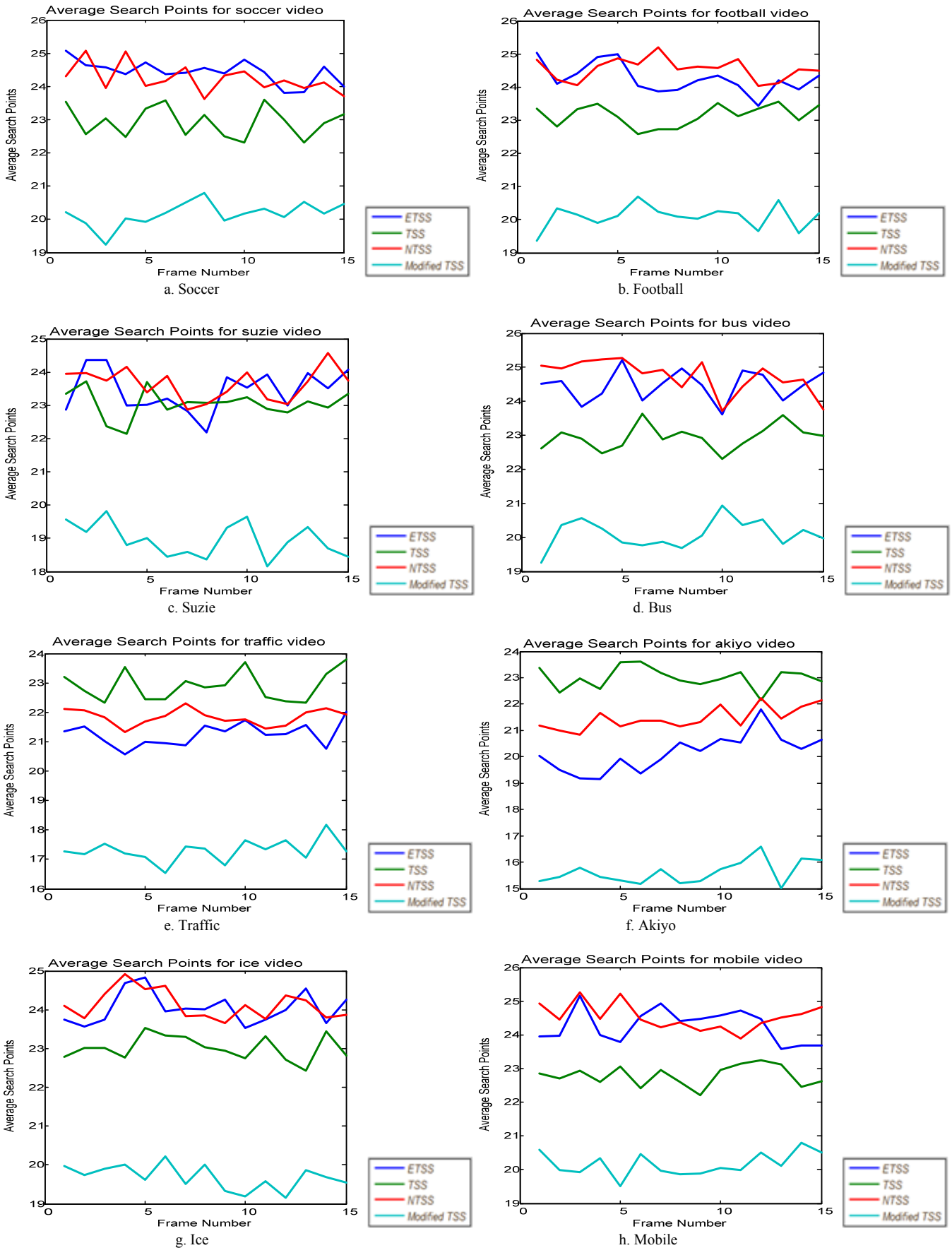
a. Soccer b. Football

c. Suzie d. Bus

e. Traffic f. Akiyo

g. Ice h. Mobile

Fig.11.Performance comparison of MTSS in terms of mean absolute difference (MAD).



a. Soccer b. Football c. Suzie d. Bus e. Traffic f. Akiyo g. Ice h. Mobile

Fig.12.Performance comparison of MTSS in terms of average search points/check points.

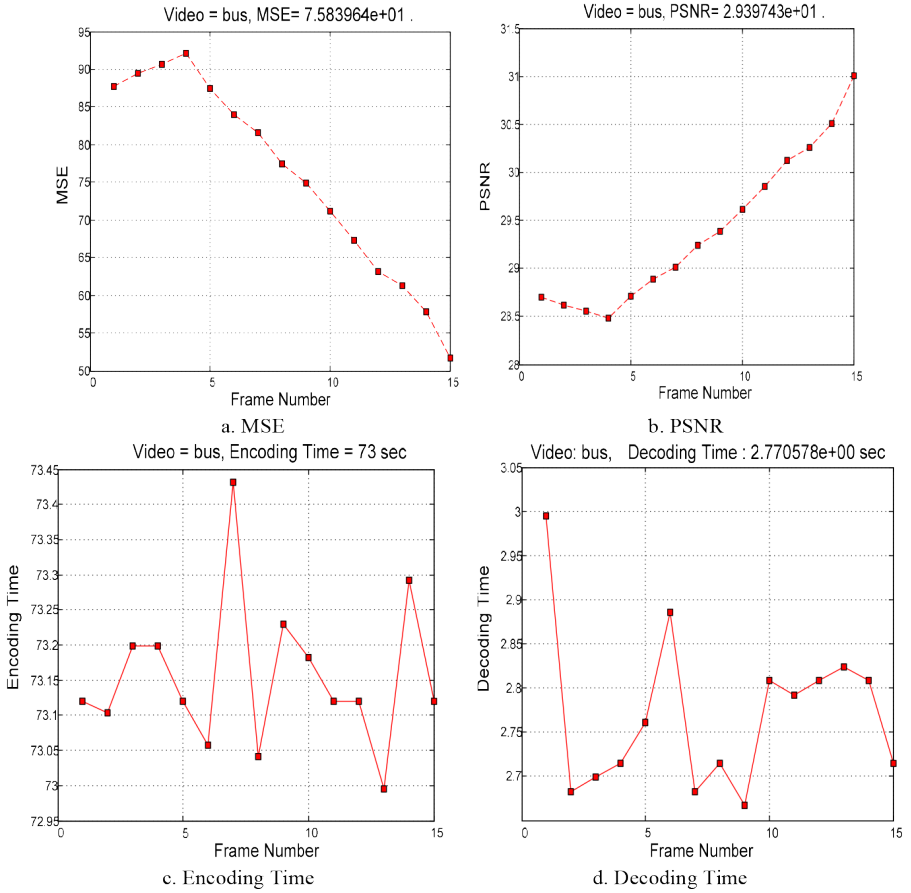


Fig.13. Performance of MTSS and WFA coding in terms of MSE, PSNR, Encoding and decoding Time.

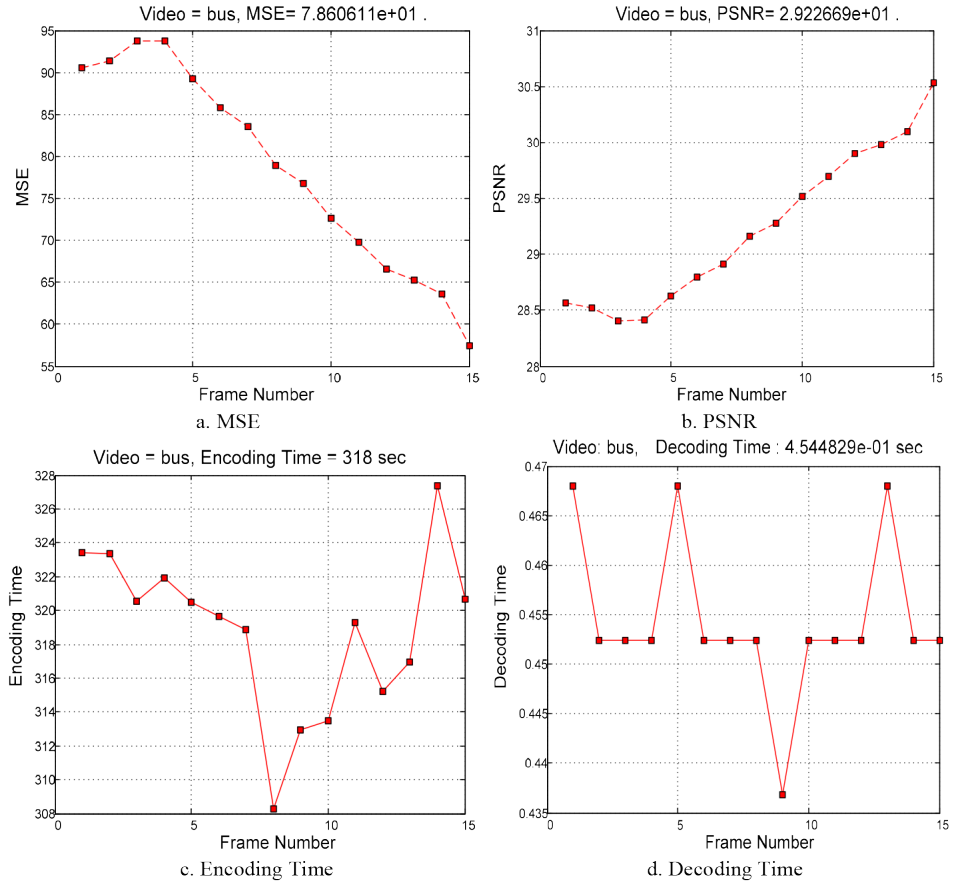


Fig.14. Performance of MTSS and Fractal Coding results for the first fifteen frames of "bus" sequences

During experimentation the search parameter p=3 pixels in both the horizontal and vertical directions for the macro block of size 16×16 are considered as the search area for a macro block.

From These Figures, the proposed MTSS coding gives less number of MAD and search points/check points required for each frame than TSS, NTSS and ETSS. The matching of one macro block with other macro block is based on the minimum output of cost function given in equation 2.

D. Evaluation and Comparison of MTSS Block Matching Estimation and Weighted Finite Automata Coding

The performance of proposed approach measured on the evaluation parameters are quality of decoded video i.e. PSNR, MSE, compression ratio, decoding and encoding time at search parameter p=3 as shown in Table II.

Fig. 13 shows the performance measure of first 15 frames of bus sequence on MSE, PSNR, encoding time and decoding time.

E. Evaluation and Comparison of MTSS Block Matching Estimation and Fractal Coding

In every image, there is a scope to have the self similarity. Fractal coding explores the self similarity existing in the images. For fractal encoding and decoding process the approach with range block of size 8×8 and domain block of size 16×16 is used. Table II to III gives the statistical performance comparison of first 15 frames of standard sequences. The proposed MTSS block matching algorithm and WFA coding approach is compared with MTSS and fractal coding approach given in Table III. Also both the approaches are compared with the frame by frame fractal coding approach. The fractal image compression approach is further implemented on image/video sequences based on intra-frame coding for comparison in terms of encoding time, compression ratio, and quality of video i.e. PSNR given in Table IV. Fig. 14 shows the performance measure in terms of MSE, PSNR and encoding time of first 15 frames of bus sequence using MTSS and fractal coding approach, and frame by frame based fractal coding approach respectively. Fig. 15 shows the 15th decoded frame of bus sequence in MTSS and WFA coding.

F. Evaluation and Comparison of Fractal Coding

From the Table II to Table IV, we can find that the proposed video compression using MTSS and WFA coding performs better than MTSS and fractal coding, and frame by frame fractal coding in terms of achieving reduced encoding time and better quality of video. The performance measure of simple fractal coding (individual frame coding) for first 15 frames of video sequences in terms of MSE, PSNR, encoding time and decoding time is given in Table IV.

Fig.15. 15th decoded frame "bus" sequences.

TABLE II

PERFORMANCE COMPARISONS OF MTSS AND WFA CODING FOR FIRST 15 IMAGE SEQUENCES AT SEARCHING PARAMETER P=3: AVERAGE MSE, PSNR, ENCODING AND DECODING TIME

Sequences	MSE	PSNR dB	Encoding Time (s)	Decoding Time (s)	Compression ratio
Soccer	51.59	31.02	71	2.739	0.3025
Suzie	13.95	36.97	72	2.793	0.2779
Traffic	12.21	37.26	73	2.858	0.3825
Bus	75.83	29.39	73	2.770	0.3447
Football	60.27	30.33	72	2.747	0.3375
Ice	43.65	31.82	66	2.759	0.2689
Akiyo	7.94	39.14	73	2.831	0.0715
Paris	22.76	34.59	69	2.711	0.902
Mobile	40.88	32.06	65	2.760	0.095

TABLE III

PERFORMANCE COMPARISONS OF MTSS AND FRACTAL CODING FOR FIRST 15TH IMAGE SEQUENCES AT SEARCHING PARAMETER P=3: AVERAGE MSE, PSNR, ENCODING AND DECODING TIME

Sequences	MSE	PSNR dB	Encoding Time (s)	Decoding Time (s)	Compression ratio
Soccer	58.55	30.46	306	0.4451	0.3052
Suzie	23.95	34.35	312	0.430	0.2783
Traffic	14.17	36.62	263	0.400	0.3839
Bus	78.60	29.22	318	0.454	0.3418
Football	63.22	30.12	324	0.433	0.3416
Ice	52.69	30.98	305	0.426	0.2720
Akiyo	16.67	35.96	250	0.401	0.0721
Paris	39.86	32.18	303	0.404	0.0895
Mobile	54.95	30.73	337	0.427	0.1009

TABLE IV

RESULTS OF FRACTAL CODING FOR FIRST 15TH FRAMES ON STANDARD DATABASES: AVERAGE MSE, PSNR, ENCODING TIME AND DECODING TIME

Sequences	MSE	PSNR dB	Encoding Time (s)	Decoding Time (s)	Compression ratio
Soccer	44.65	31.64	4332	0.411	0.2848
Suzie	27.63	33.71	4330	0.408	0.2493
Traffic	27.05	33.81	4324	0.410	0.3612
Bus	58.88	30.43	4345	0.430	0.1410
Football	65.46	29.97	4349	0.416	0.2412
Ice	50.03	31.14	4311	0.409	0.2523
Akiyo	34.26	32.78	4328	0.407	0.1817

VI. CONCLUSION AND FUTURE SCOPE

In this paper, a modified three step search (MTSS) algorithm and WFA coding approach is proposed to reduce the encoding time. A MTSS block matching motion estimation approach performs better in terms of small MAD and less average search points/check points required than the TSS, NTSS and ETSS at search parameter p=3 pixels. MTSS performs efficiently for the frames with slow and fast motions. Hence, the MTSS algorithm is suitable for video applications. The proposed MTSS and WFA coding approach performed better than MTSS and FC as well as frame-by-frame FC in terms of the encoding time. In MTSS and WFA, the encoding time is reduced by 70% to 80%

in comparison with MTSS and FC.

Developed block matching algorithms have scope for improvement through optimization of searching process by exploring the nearest neighborhood of pixels. The developed block matching algorithm can be combined with different coding mechanisms from video compression point of view.

References

[1] Gonzalez, R. C., Woods, R. E. (2005) '*Digital Image Processing*', Second Edition, Pearson Education Asia.

[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] Zhu, S., Tian, J., Shen, X. and Belloulata, K. (2009) '*A new cross-diamond search algorithm for fast block motion estimation*', In the IEEE Int. Conf. Image Processing, ICIP'09, Cairo, Egypt, Vol. I, pp. 1581-1584.

[6] Zhu, S., Tian, J., Shen, X. and Belloulata, K. (2009) '*A Novel Cross-Hexagon Search Algorithm Based on Motion Vector Field Prediction*', In the IEEE Int. Symposium on Industrial Electronics, ISIE'09, Seoul, Korea, pp. 1870-1874.

[7] Zhu, S., Hou, Y., Wang, Z. and Belloulata, K. (2010) '*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, Vol. 8, pp.360-364.

[8] Koga, T., Iinuma, K., Hirano, A., Iijima, Y. and Ishiguro, T. (1981) '*Motion compensated interframe coding for video conferencing*', In Proc. National Telecommunications Conf., New Orleans, LA, pp. G5.3.1 –G5.3.5.

[9] Jain, J. R. and Jain, A. K. (1981) '*Displacement measurement and its application in interframe image coding*', IEEE Transactions on Communications, vol. 29, pp. 1799-1808.

[10] Puri, A., Hang, H. M. and Schilling, D. L. (1987) '*An efficient block matching algorithm for motion compensated coding*', Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc., pp. 1063-1066.

[11] Ghanbar, M. (1990) '*The cross search algorithm for motion estimation*', IEEE Trans. Commun., Vol. COM-38, pp. 950-953.

[12] Urabe, T., Afzal, H., Ho, G., Pancha, P. and Zarki, M. E. (1994) '*MPEG Tool- an X window-based MPEG encoder and statistical tool*', Multimedia a syst. 1(5). pp. 220-229.

[13] R. Li, Zeng, B. and Liou, M. L. (1994) '*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.

[14] Po, L. -M. and Ma, W.-C. (1996) '*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.

[15] Liu, L. K. and Feig, E. (1996) '*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.

[16] Zhu, S. and Ma, K.K.(2000) '*A new diamond search algorithm for fast block-matching motion estimation*', IEEE Transactions on Image Processing, vol. 9, no. 2, pp. 287-290.

[17] Cheung, C. H. and Po, L. M. (2002) '*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.

[18] Jing, X. and Lap-Pui, C.(2004) '*An efficient three-step search algorithm for block motion estimation*', IEEE transactions on multimedia, Vol. 6, No.3, pp. 435-438.

[19] Tham, J.Y., Ranganath, S., Ranganath, M. and Kassim, A. A.(1998) '*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.

[20] Chen, H. M., Chen, P. H., Yeh, K. L., Fang, W. H., Shie, M. C. and Lai, F.(2007) '*Center of Mass-Based Adaptive Fast Block Motion Estimation*', EURASIP Journal on Image and Video Processing, vol. Article ID 65242, 11 pages.

[21] Zhu, C., Lin, X. and Chau, L-P.(2002) '*Hexagon-based search pattern for fast block motion estimation*', IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, pp. 349-355.

[22] Cheung, C. H. and Po, L. M. (2000) '*Normalized partial distortion search algorithm for block motion estimation*', IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, pp. 417-422.

[23] Li H., Liu M. (2009) '*Cross-Hexagon-based motion estimation algorithm using motion vector adaptive search technique*', International Conference on Wireless Communications & Signal Processing, Nanjing, 2009, pp. 1-4.

[24] Belloulata, K., Zhu, S., and Wang, Z. (2011) '*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.

[25] Acharjee, S., Ray, R., Chakraborty, S., Nath, S., & Dey, N. (2014), "Watermarking in motion vector for security enhancement of medical videos", IEEE *International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014,* pp. 532-537.

[26] Ikeda, N., Araki, T., Dey, N., Bose, S., Shafique, S., El-Baz, A., ... & Suri, J. S. (2014), "Automated and accurate carotid bulb detection, its verification and validation in low quality frozen frames and motion video", *International angiology: a journal of the International Union of Angiology*, *33*(6), 573-589.

[27] Dey, N., Ashour, A., & Acharjee, S. (2017), "Applied Video Processing in Surveillance and Monitoring Systems", pp. 1-321, Hershey, PA: IGI Global. doi:10.4018/978-1-5225-1022-2.

[28] Kamble, S. D., Thakur, N.V., Malik, L. G. and Bajaj, P. R. (2014) '*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.

[29] Culik, K. and Kari, J. (1995) '*Inference algorithms for WFA and image compression*', In Y. Fisher, editor, Fractal Image Compression, chapter 13, pages 243-258. Springer-Verlag.

[30] Kari, J. and Franti, P. (1994) '*Arithmetic coding of weighted finite automata*', Theoretical Informatics and Applications, 28(3-4):343-360.

[31] Hafner, U.(1996) '*Refining Image Compression with Weighted Finite Automata*', IEEE Data compression Conference, pp. 359-368.

[32] Katritzke, F. (2001) '*Refinements of Data Compression Using Weighted Finite Automata*', Ph.D. dissertation.

[33] Katritzke, F., Merzenich, W., and Thomas, M. (2003) '*Enhancements of partitioning techniques for image compression using weighted finite automata*', Elsevier, 2003.

[34] Kamble, S. D., Thakur, N.V., Malik, L. G. and Bajaj, P. R. (2015) '*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.

Shailesh D. Kamble received Bachelor of Engineering degree in Computer Technology from Yeshwantrao Chavan College of Enginering, 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. Raisoni 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 Enginering, 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. Raisoni 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.