

Pattern Based Motion Estimation using Zero Motion Pre-judgement and Quantum behaved Particle Swarm Optimization

Yogananda Patnaik¹ and Dipti Patra², Non-members

ABSTRACT

Motion estimation is a fundamental and resource hungry operation in most of the video coding applications. The most popular method used in any video coding application is block matching motion estimation (BMME). This conventional fast motion estimation algorithm adopts a monotonic error surface for faster computation. However, these search techniques may trap at local minima resulting in erroneous motion estimation. To overcome this issue, various evolutionary swarm intelligence based algorithms were proposed. In this paper, a pattern based motion estimation using zero motion prejudgment and Quantum behaved Particle Swarm Optimization (QPSO) algorithm is proposed, referred to as the Pattern Based Motion Estimation (PBME) algorithm. The notion of QPSO improves the diversity in the search space, which enhances the search efficiency and helps in reduction of the computational burden. At the same time, QPSO needs fewer parameters to control. Therefore, the proposed algorithm enhances the estimation accuracy. An initial search pattern (Hexagonal Based Search) was used which speeds the convergence rate of the algorithm. From the simulation results, it was found that the proposed method outperformed the existing fast block matching (BMA) algorithms of the search point reduction by 40–75%.

Keywords: PSO; QPSO; Motion Estimation; Motion Vector (MV)

1. INTRODUCTION

The most crucial component of any block based video coding system is motion estimation. This is mainly due to the use of temporal redundancy between progressive frames of video [1]. Consecutive frames of a video sequence have high correlation. So high coding efficiency can be achieved in any video coding system, with reduction in temporal redundancy. Exploitation of temporal redundancy between successive frames in a video codec is possible

with a motion compensation technique by predicting the current frame from the past one. Motion estimation (ME) has a vital role in inter-frame prediction. From the available motion estimation techniques, the block-matching algorithm (BMA) is considered the most popular because of its simplicity and has been adopted in most of the video coding standards, namely H.264/AVC and H.265/HEVC [2]. In this case, BMA frames are broken down to blocks, and individual motion vectors are calculated for each block. Here, for each block in the current frame, the motion estimation process is applied and the motion vector is the best matching block in the reference frame. This best matching block becomes the predictor for the current block. The most accurate and computationally expensive method is a full search or an exhaustive search because block matching is done a on block by block basis for each and every block [3]. Hence, there is an urgent need for reduction in the computational load while maintaining the quality of the achieved motion information, as well as to find an optimal solution by calculating all possible candidate blocks in the reference frame within its search area.

The application areas of video coding includes fixed and mobile telephony, video conferencing applications, DVD and HDTV applications [4]. In the case of Block Based Motion Estimation (BBME), frames are divided into non-overlapping blocks, and each block of the current frame is matched block by block and connected to a search window in the reference frame based on some minimum matching criteria. This can be either a mean, an absolute difference (MAD) or a sum of absolute difference (SAD), among other methods. The BBME can be regarded as a problem of optimization. The full search algorithm gives a definite optimal solution because it searches block by block. This method cannot be used in a practical scenario because of its large computational burden. To overcome this limitation, various fast search techniques have been proposed in the literature for several video coding standards such as the H.26x series, MPEG series, and HEVC. The vital parameters to develop ME algorithms are the search pattern, choice of an initial centre and search strategy. These three factors are used for the calculation of machine performance and the peak signal to noise ratio (PSNR) [1] of the algorithm.

Manuscript received on April 22, 2017 ; revised on May 15, 2017.

Final manuscript received on May 30, 2017.

^{1,2} The authors are with DDepartment of Electrical Engineering National Institute of Technology Rourkela, India, E-mail: ypatnaiknitrkl@gmail.com and dpatra@nitrkl.ac.in

2. RELATED WORKS

To fulfill real-time processing needs in multimedia applications, the speed of ME algorithms needs to be increased. Therefore, faster motion estimation search methods became popular and have been reported in the literature, where the research is still aimed at reducing the computational costs of full-search algorithms. Fast motion estimation methods include the three-step search (TSS) [5], four-step search (FSS), N-step search (NSS) [6], diamond search (DS) [7], cross diamond search methods (CDS) [8], adaptive rood pattern search (ARPS) [9], and hexagonal search (HEXPBS) [10], among others.

These search techniques will be liable to entrapment at local optima on the error surface. The basic hypothesis behind these search methods comes from the concept that the block distortion measure is reduced monotonously if the search points move from the furthest point towards the optimum point. Hence, these fast search methods get trapped at local best solutions instead of achieving the global optimum. To overcome the problem of local minima, various approaches have been proposed in the literature for attaining global optimization to overcome the difficulties of motion estimation. Among available methods, genetic algorithms are often considered. These algorithms are more complex and computationally expensive. Simulated annealing is also applied adaptively of the search process by choosing an intense search region. Other algorithms such as the ant bee colony [11] and different evolution algorithms have also been proposed for motion estimation. However, few attempts have been made in the literature to use of particle swarm optimization (PSO) for solving the problem of motion estimation [3, 12, 13, 14]. However, these methods have intense computational complexity and low estimation accuracy as compared to more traditional methods. To increase the speed of the traditional PSO algorithm presented in [3], the starting point of the particles used a predetermined pattern instead of a random one. The initialization of the particles was either a square or diamond shaped pattern about the centre.

A modified PSO algorithm was proposed in [1] to fulfill the requirement for a low computational load while preserving high motion estimation accuracy. To speed the analysis and simultaneously increase the motion estimation accuracy, various schemes were implemented and applied to the motion estimation process. The modified PSO with certain strategies produced remarkable improvements regarding both estimation accuracy and computational efficiency over other state-of-art methods. However, this can be further increased by using a more accurate block matching process that can further lower the computational complexity. One of the improved PSO methods for higher accuracy and low computational complexity is QPSO. This paper deals with pattern based motion

estimation techniques using QPSO and zero motion prejudgment. QPSO is used here because a particle remains in the bound state, which can appear at any point in the entire search space with a definite probability, even if the position is far away from the learning inclination point.

This QPSO algorithm outperforms the conventional PSO algorithm due its ability to attain global convergence and its quantum system has more states than a linear system. In the proposed method, initially, a video sequence is given as input followed by extraction of frames. Each frame is divided into non-overlapping blocks. Before doing motion estimation, zero motion prejudgment is performed as a pre-processing step to check whether the blocks are static or in motion. If the blocks are static, then there is no necessity to do motion estimation. This step reduces computation and saves memory. Then, a predefined or fixed search pattern, namely a hexagonal search pattern, is used for distribution of the search in the search window. The QPSO algorithm is applied for a faster search in motion estimation. Here, P_{best} and G_{best} are calculated. The proposed method saves memory and simultaneously speeds up the process substantially.

The rest of this paper is organized as follows: Section 2 renders materials and methods which discuss the background of basic motion estimation techniques, as well as evolutionary algorithms such as PSO and QPSO. It also reviews some of the state-of-art regarding the evolutionary approach towards motion estimation. Section 3 focuses on the proposed method for motion estimation using QPSO and zero motion prejudgement. The proposed technique is validated and verified using various performance measures and metrics such as Peak Signal to Noise Ratio (PSNR), Mean Structural Similarity Index (MSSIM), Feature Similarity Index (FSIM) and Universal Quality Index (UQI) which are discussed in Section 4. Finally, the concluding remarks are presented in the Section 5.

The major contributions of this paper are:

- Motion estimation using an evolutionary approach
- Zero motion prejudgement to find out the blocks that are stationary and in motion
- Use of the QPSO algorithm with a hexagonal based pattern for finding a motion vector in a motion field with fewer search points

3. MATERIALS AND METHODS

3.1 Motion Estimation

Motion estimation is applied in most video application scenarios, namely video compression, segmentation and tracking. The most popular and widely accepted algorithm for motion estimation is the block-matching algorithm (BMA) [15]. In the case of block

matching, a frame is broken down into various rectangular blocks and motion vectors for all the blocks are estimated on a block by block basis over a search range in the reference frame. This is done by calculating the nearest matching block of pixels as per the defined matching criteria, primarily the sum of squared differences (SSD) or sum of absolute differences (SAD). Motion estimation can achieve a substantial amount of compression by exploiting the temporal redundancy that exists in most video sequences.

Various motion estimation methods have been discussed in the literature. They strive for reduction of computational complexity. These methods are the block-matching algorithm, parametric based models [16], optical flow models [17] and pel recursive techniques [18]. Block-matching algorithms seem to be the most popular one among those available. This is because of their effectiveness and simplicity for both hardware and software applications. The predictor block is the best matching block where the displacement is given by a translational motion vector (MV). Hence block-matching can be treated as an optimization problem to find the best matching block in a predefined search space. The full search algorithm [19] provides an optimal solution on minima matching error because this algorithm takes all the candidate blocks at one time. However, the computational burden is increased substantially, which restricts its use in real-time video applications. The sum of absolute difference can be calculated as

$$SAD_{(u,v)} = \sum_{i=1}^N \sum_{j=1}^N |f_t(i,j) - f_{t-1}(i+u,j+v)| \quad (1)$$

where $-w \leq u, v \leq w$. (u, v) is the distance of the candidate block to the current block, $f_t(i, j)$ implies the Engineering design, and many more [22].

Gray value of $(i, j)^{th}$ pixel in the current frame, and $f_{t-1}(i, j)$ shows the gray value of $(i, j)^{th}$ pixel in the previous frame. The displacement vector (u, v) of the candidate block with the minimum $SAD(u, v)$ gives the motion vector. The size of block is $N \times N$ and u, v are in the range $[-w, w]$. The basic H.264 codec structure is shown in Fig. 1. The basic motion estimation process using a block matching algorithm is illustrated in Fig. 2.

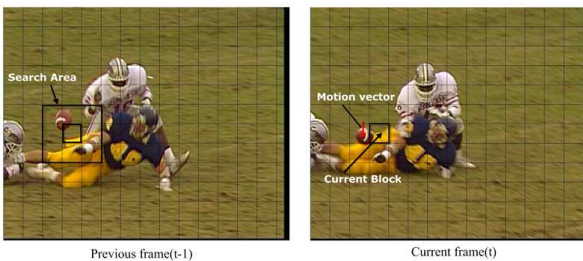


Fig.2: Block Matching Process.

3.2 Overview of QPSO

In a PSO system, the search area cannot cover the entire viable region, and global convergence is not assured. This is the important cause of early termination in PSO. To overcome this issue, quantum-behaved particle swarm optimization (QPSO) is proposed [20]. In the QPSO algorithm, the position of a particle is portrayed by its local attractor and by a probability density function. By adopting this approach, there is no constraint on particle trajectory. Furthermore, there is only a single parameter in QPSO. Recently, QPSO was successfully employed in a variety of optimization problems, such as constrained optimization, multi-objective optimization, and engineering design, among many others [22].

QPSO, a probabilistic algorithm, was proposed by Sun *et al.* [23]. The details of the QPSO algorithm are depicted in Algorithm 1. The source of inspiration in QPSO algorithm was primarily the quantum mechanics of particles and the trajectory analysis of PSO, which was formulated by Clerk and Kennedy in 2002. From the trajectory analysis of PSO, it can be observed that every particle in a PSO algorithm oscillates around and converges at a local minimum or remains in a bound state. In the case of QPSO, the particles have persistent quantum behaviour and are in a bound state. Furthermore, it was also assumed that the particles were attracted towards a quantum potential well that is centred on its local point giving rise to a stochastic update equation. Later, a mean best position was introduced to the algorithm to increase the global search capacity of this QPSO [23]. The flowchart of the QPSO algorithm is shown in Fig. 3. Updating the position of particles is done as follows:

$$X_{k,n+1}^l = p_{k,n}^l \pm \frac{L_{k,n}^l}{2} \ln\left(\frac{1}{u_{k,n}^l}\right) \quad (2)$$

where $X_{k,n+1}^l$ is the current position vector at $(n+1)^{th}$ iteration, $u_{k,n}^l$ is a random number which is uniformly distributed in $(0,1)$ and $p_{k,n}^l$ [24] is a local attractor that ranges from $p_{k,n}^1, p_{k,n}^2, \dots, p_{k,n}^N$

$$p_{k,n}^l = \varphi_{k,n}^l P_{k,n}^l + (1 - \varphi_{k,n}^l) P_{g,n}^l \quad (3)$$

and $p_{k,n}^l$ is the local best previous position which is a vector that ranges from $p_{k,n}^l = p_{k,n}^1, p_{k,n}^2, \dots, p_{k,n}^N$.

$$L_{k,n}^l = 2\beta |p_{k,n}^l - X_{k,n}^l| \quad (4)$$

where $\varphi_{k,n}^l$ is uniformly distributed random number in $(0,1)$ also:

$$X_{k,n+1}^l = p_{k,n}^l \pm \beta |p_{k,n}^l - X_{k,n}^l| \ln\left(\frac{1}{u_{k,n}^l}\right) \quad (5)$$

where β being the contraction-expansion (CE) coef-

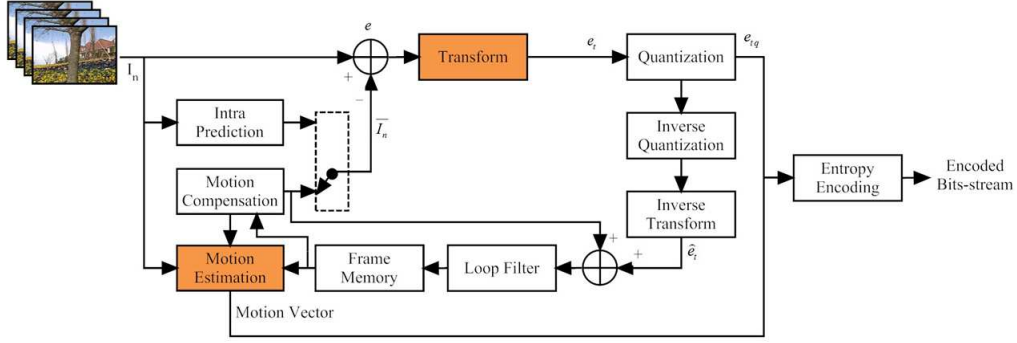


Fig.1: Basic H.264 Codec structure.

Algorithm 1: QPSO Algorithm

begin

Initialization of actual positions and the P_{beat} positions of all particles; Evaluation of their fitness value and global best positions G_0 and Set $n = 0$;

while the end point condition is not satisfied **do**

choose an appropriate value for α ;

for $i = 1$ to M **do**

Objective function value evaluation $f(X_i, n)$;

Update $P_{i,n}, n$ and G_{ni} ;

for $j = 1$ to N **do**

$\sigma_{i,n} = rand1(\cdot)$

$P_{i,n} = \varphi_{i,n} p_{i,n} + (1 - \varphi_{i,n}) g_n$

$u_{i,n+1} = rand2(\cdot)$;

if $rand3(\cdot) < 0.5$ **then**

$X_{i,n+1} = P_{i,n} + \alpha |X_{i,n} - p_{i,n}| \ln(1/u_{i,n+1})$

else

$X_{i,n+1} = p_{i,n} - \alpha |X_{i,n} - p_{i,n}| \ln(1/u_{i,n+1})$

end

end

end

end

Set $n = n + 1$;

end

end

ficient that is used as a tuning factor for convergence speed of the algorithm.

The particle positioning is updated according to the equation:

$$X_{k,n+1}^l = p_{k,n}^l \pm \beta |P_{k,n}^l - X_{k,n}^l| \ln\left(\frac{1}{u_{k,n}^l}\right) \quad (6)$$

Finally, the QPSO equation is given as:

$$X_{k,n+1}^l = p_{k,n}^l \pm \beta |mbest_n^l - X_{k,n}^l| \ln\left(\frac{1}{u_{k,n}^l}\right) \quad (7)$$

the range of β is set as 1 and is gradually reduced to 0.5 linearly. The method for choosing this β value is by trial and error. In our case, the most suitable value of β was found to be 0.5 for better performance

of the algorithm.

4. PROPOSED METHOD

The proposed method deals with a pattern based motion estimation technique using QPSO and zero motion prejudgment. The flow chart of the proposed algorithm is given in Fig. 5. Here, initially a video sequence is given as input followed by extraction of frames. Then, these individual frames are partitioned into non-overlapping blocks. The images are partitioned into blocks of 8x8 pixels that are called as macroblocks. The macroblock partition is hierarchical on the block position, and therefore every macroblock consists of four blocks.

Before doing motion estimation, zero motion prejudgment is performed as a preprocessing step to check whether the blocks are static or in motion. If

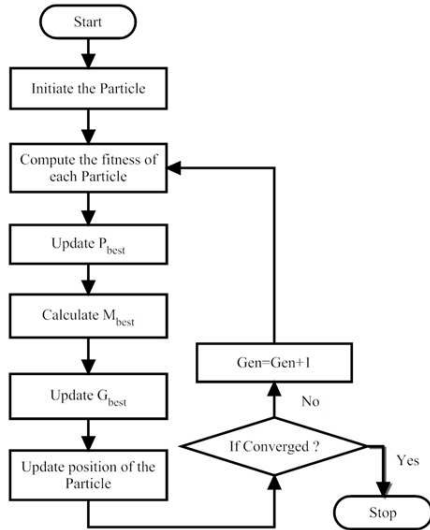


Fig.3: Flowchart for QPSO Algorithm.

the blocks are static, then there is no need to do motion estimation. This step reduces computation and saves memory. After the non-static blocks are considered, a predefined search pattern, namely a hexagonal search pattern, is used for the search in the search window. Then, the QPSO algorithm is applied for a faster motion estimation process. Here, P_{best} and G_{best} are calculated.

The block diagram of the proposed method is shown in Fig. 4 and generally consists of four parts:

1. Pre-processing
2. Zero motion pre-judgement
3. Choice of fixed pattern
4. Efficient search using QPSO

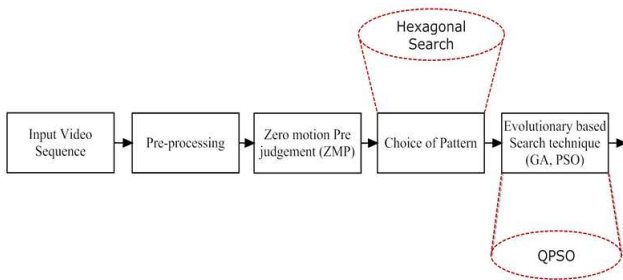


Fig.4: Proposed method using QPSO.

4.1 Pre-Processing

The pre-processing step consists of the following steps:

1. Conversion of input video to frames
2. Conversion of colour images to a gray scale
3. Resizing the image frames
4. Block partitioning

The first step is conversion of the input video data into a number of frames. These frames are converted

to gray scale images. Then the frames are resized as necessity of upsampling or downsampling. These resized frames are converted into blocks using block partitioning, which is done by converting all of the frames to 8×8 blocks.

4.2 Zero motion pre-judgement

Successive video frames in a sequence consists of around 70% static macro blocks that do not require any further search. So, to calculate the static macro blocks, the zero motion pre-judgement (ZMP) mechanism is used before the start of the actual motion estimation process. Therefore, a considerable reduction in computational effort is possible while the remaining search becomes faster and thus saves memory. To check if a block is stationary or not, block distortion is measured and compared with a predefined threshold value, T . If the distortion value is below the threshold, then it is a stationary block and the search process is stopped. Thus, its resultant motion vector is $(0, 0)$. The detailed process of determining the threshold value is given in the literature [25].

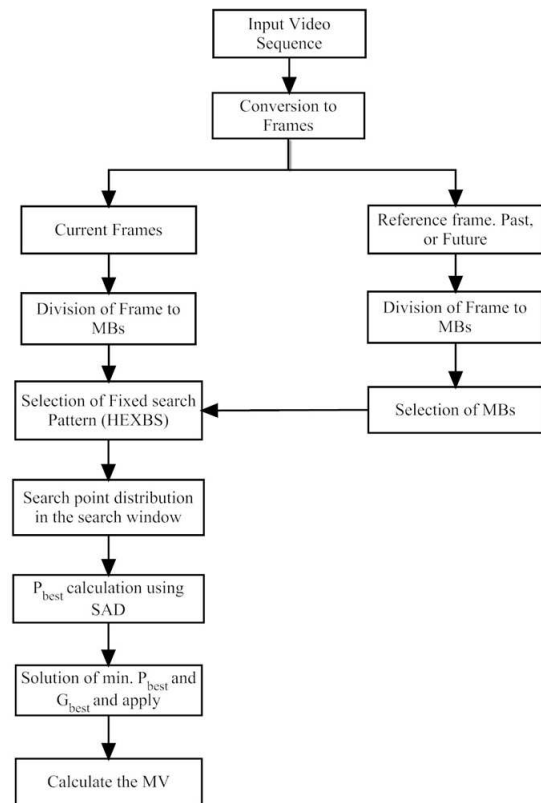


Fig.5: Flowchart of the proposed algorithm

4.3 Choice of Fixed Pattern: Hexagonal Based search

An assumption made by Zhu et al. [10] is that when the distortion within a small neighbourhood increases monotonically, a circular shaped search

Algorithm 2: Proposed PBME using Zero motion Pre-judgement and QPSO

```

begin
  Initialization of actual positions and the  $P_{best}$  positions of all particles;
  for every frame  $i$  do
    for every macro block  $j$  do
       $zmpc \leftarrow SAD(I_{i-1}(j), I_i(j));$ 
      if  $zmpc < t$  then
        the macro block (MB) is static
         $MV = [0, 0]$ 
        Continue
      else if  $MB_j$  is in the leftmost column of frame  $i$  then
        if  $MB_j == 1$  then
          initial particles are in square pattern
        else if  $MB_j$  is in the bottom right column then
          initial particles are in diamond pattern
        else
           $MB_j$  is in the bottom right corner column initial particles are in cross
          diamond pattern
        end
      else
        initial particles are in hexagonal pattern
      end
    end
  end
  start the QPSO search
  for each iteration time  $t$  do
    for each particle  $p$  do
      Evaluate the SAD using equation 1
      update  $P_{best}$  and  $G_{best}$ 
      update position
    end
  end
end
end
end

```

method is more effective for obtaining the highest uniform search speed [27]. A triangle or diamond shape cannot be approximated by a circle, so a hexagon can be used. This pattern consists of seven points used in a search. Of these, the centre point is enclosed by the other six points of a hexagon. Among these six points, two horizontal points are at a distance of two units from the centre and the remaining four points are at a distance of $p=5$ from the centre. These six end points are uniformly distributed around the centre. Each time the centre point moves in relation to any of the six points, then that point results in three new points, while the remaining three points overlap. This search process continues until it comes to an edge. The hexagonal search pattern and the motion estimation direction is shown in Fig. 7

4.4 Efficient search using QPSO

A QPSO based approach is an efficient search technique compared to the conventional rapid search techniques because a particle remains in a bound state,

Table 1: Video Data Set Configuration.

Sequence	Y_RES	UV_RES	Freq.	N F	CF	Class
Silent	720×576	360×576	50	300	4:2:2	A
News	720×486	360×486	60	300	4:2:2	A
Akiyo	720×486	360×486	60	300	4:2:2	A
Container	720×486	360×486	60	300	4:2:2	B
Hall Object	720×486	360×486	60	300	4:2:2	A
Mother Daughter	720×480	360×240	60	300	4:2:0	A
Foreman	720×576	360×576	50	300	4:2:2	B
Silent	720×576	360×576	50	300	4:2:2	A
Coastguard	720×486	360×486	60	300	4:2:2	B

which can be present at any point in the entire search space with a definite probability, even if the position is far away from the learning inclination point. Thus, in a quantum system, the principle of superposition holds, and this system has far more states than a linear system. Also, in a quantum system, a particle can be present at any point with a certain probability distribution as it has no fixed trajectory [28]. Hence, the QPSO algorithm has a global convergence property and therefore reduces the computational load result-



Fig.6: Video data set (a) Garden (b) Football (c) Tennis (d) Clare [26].

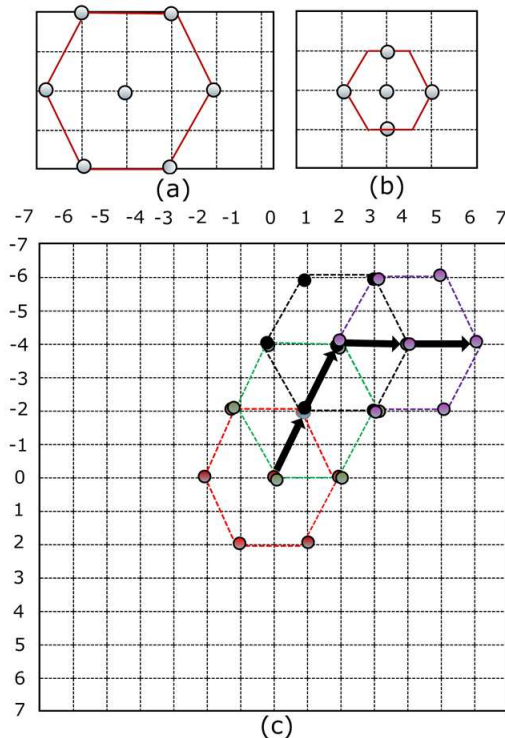


Fig.7: Hexagonal Pattern: (a) Large Hexagonal pattern; (b) Small hexagonal pattern; and (c) Example of a search path locating the motion vector (+4,-4).

ing in an efficient search.

5. SIMULATION RESULTS

This section gives a detailed discussion of the various evaluation parameters the were used to quantify the effectiveness of the proposed pattern based QPSO motion estimation algorithm. A detailed comparative performance of the proposed algorithm with the state-of-the-art is presented in this section.

5.1 Experimental Environment

This section deals with the simulation results of the proposed QPSO based ME compared to those of the existing conventional algorithms. The simulations were done with the standard benchmark datasets as

shown in Table 1. These standard videos have different format degrees and types of motion. They are QCIF: 176×144 , CIF: 352×288 and SIF: 352×240 . The Clare video is gentle, smooth with low motion variation, which consist of mostly stationary blocks. Whereas the Tennis video is complex and contains a medium level of motion. However, sequences like Garden and Football consist of high motion activities, which are dependent on camera panning and complex motions.

5.2 Experimental results and discussion

The input individual frames were partitioned into macroblocks of size 8×8 with a maximum displacement within a search range of ± 7 pixels. For the performance comparison of the proposed method, various conventional search algorithms were implemented. The motion displacement 'p' or the search range has a direct impact on both computation complexity and prediction quality of the block matching technique. A small p provides poor compensation for areas with faster motion and thus resulting in poor prediction quality. Alternatively, a large p results in good prediction quality, thereby increasing the computational load since there are $(2p+1)^2$ blocks. A large p will produce a longer MV, thus increasing the motion overhead. So, in general, the maximum permissible displacement of $p = \pm 7$ pixels is sufficient for low bit rate environment.

Table 2: Average PSNR of different methods with various data sets.

Test VS	FS	NTSS	DS	ARPS	PSO	QPSO
Tennis	25.6	23.85	23.63	23.62	24.68	25.02
Football	18.7	17.53	17.31	17.41	17.91	18.25
Clare	39.8	38.89	38.89	38.88	39.63	39.71
Garden	20.2	18.83	18.24	18.79	19.26	19.51

The simulations were based on performance indices such as PSNR, SSIM, and the number of search points. The simulations were performed on a core i7 processor at 3.7 GHz with 4GB RAM. To compare the performance of the proposed method with the standard methods, i.e., FS, TSS, 4SS, ARPS and DS, were implemented. The average value of the PSNR and search points were calculated to provide the statistical significance of the proposed method. As this is an iterative process, the permissible total number of iterations (It_{max}) has to be limited. The simulation results were done with an initial particle position of a hexagonal based pattern. Table 2 illustrates the average PSNR value of a video frame obtained using various methods. From this table, it can be seen that the FS method had the maximum PSNR value for all the data sets. Several experiments were carried out and verified with all the standard datasets, but here only four data set results are included in this report.

The search efficacy is usually calculated by finding

the number of search points used for the estimation of a motion vector. Table 3 shows that the proposed QPSO algorithm takes the smallest number of average search points, in the range of 7.23 to 9.52, as compared to other state-of-the-art techniques. The simulations were tested using all four data sets as shown in Fig. 6. A fewer number of search points reduced the computational burden which enhanced the efficacy of motion estimation thereby improving the efficiency of video coding.

It is found that PSO took 20 iterations to converge, whereas the proposed motion estimation algorithm converged after only 5 iterations. A comparison of computation time is presented in Table 6.

Table 3: Average PSNR of different methods with various data sets.

Algorithms	Garden	Clare	Football	Tennis
FS	190.5614	190.5614	190.5614	190.561
NTSS	22.1077	22.2635	22.1451	22.098
DS	11.7323	12.6424	12.6715	15.386
ARPS	10.6322	11.9778	12.7612	16.349
PSO	8.5291	9.6247	10.8664	11.051
Proposed	7.2315	8.3191	9.3214	9.529

Table 4: Average Speed Improvement Rate (SIR).

Average SIR in %	Garden	Clare	Football	Tennis
QPSO Over NTSS	67.29	62.63	57.91	56.88
QPSO Over DS	38.36	34.20	26.44	38.07
QPSO Over ARPS	31.98	30.55	26.96	41.71
QPSO Over PSO	15.21	13.57	14.22	13.78

In the proposed scheme, i.e., the QPSO based search algorithm, the state of the particles was described with the help of only a position vector instead of position and velocity. Furthermore, from the QPSO algorithm, we can see that, there is only one control parameter. This makes realization simpler than the PSO algorithm. QPSO performed better because the number of computations was reduced while maintaining the same or improved quality of the video data. This was mainly due to the mutation operator that provided diversity in the search space and thus increased its global search capability.

This technique provided perfect motion estimation with less computational complexity. It also provided high accuracy when compared to a full search and a diamond search.

Different search technique

FS: The full search algorithm is very easy to implement, and it gave the most accurate results. This is because it calculates all possible displacements within a search range using a block distortion measure. So, no specific algorithm is necessary. It is merely a 2-D search.

TSS: As three steps are used to find the best matched macroblock within the search window of the reference

frame, the name is three step search. The step size of the search window is initially set to half of the search area. Nine points, including the centre point and eight check points on the boundary of the search window, are selected at each step. Next, the search centre moves forward to the matching point with the minimum SAD of the first step and the step size of the second step is reduced by a factor of two. The stopping criteria is when the step size is reduced to a single pixel, and the optimum motion vector with the minimum SAD is obtained.

4SS: The four step search algorithm uses nine points for comparison, and then the points are selected based on the following algorithm: the search step starts with a size of 2. The MAE is calculated for all the points, and the point with minimum MAE detected. Then, the centre is moved to the detected point. This is done until it reaches to the boundary and the step size is reduced to a single pixel pixel.

ARPS: A rood-shaped search pattern is used, in which the size of the rood arms have a provision for adjustment adaptively while searching. The motion vector of the immediate left neighbouring block is used to predict the motion vector for the current block.

DS: The Diamond Search uses two search patterns, a large diamond for a general purpose gradient search and a smaller diamond for final stage improvement. The approach is very much similar to the 2-D Logarithmic search in that a large diamond pattern at stage k is centred on the point with a minimum BDM from stage $k-1$. When the search remains at the centre of the pattern, the small diamond pattern is invoked to refine the motion vector before termination.

HEXBS: The HEXBS may find motion vector in a motion field with fewer search points than the diamond search algorithm.

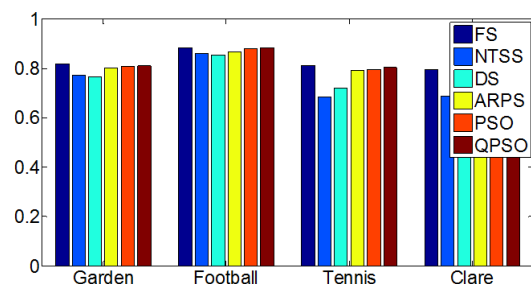


Fig. 8: SSIM plot for all the four video sequence.

The larger the motion vector the more significant the speed gain for this method. The simulation results verified the superiority of the proposed method to other fast search methods by utilizing fewer search points at the cost of minimum degradation in distortion. HEXBS also uses fewer search points each time, thereby saves cost and hardware size.

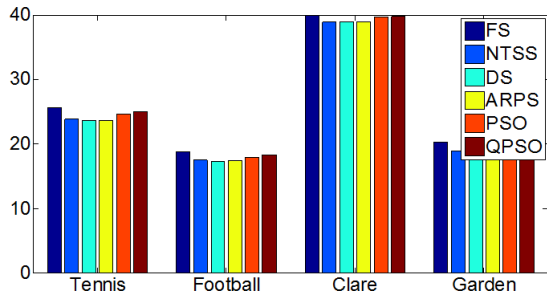


Fig. 9: Average PSNR plot for the four video sequences

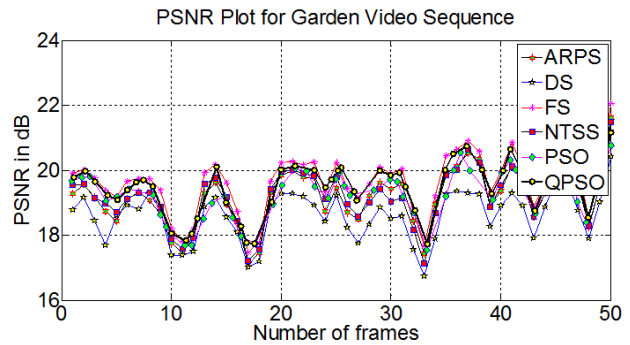


Fig. 10: PSNR plot for the Garden sequence

5.3 Comparative analysis

The average speed improvement rate (SIR) for various algorithms is summarized in Table 4. The SIR in SIR % is given by:

$$SIR = \left(\frac{N_2 - N_I}{N_1} \right) \times 100\% \quad (8)$$

where N2 is the number of search points used in method 2, while N1 is the number of search points used in method 1. The results in Table 4 demonstrate that the proposed method can reduce the number of search points from 13% to 67% compared to other block matching algorithms. Fig. 10, 11, 12, 13 display the plots for average PSNR for all four data sets on frame by frame basis. The SSIM plot is depicted in Fig. 8. This plot shows that the reconstruction quality of the frames for QPSO was better as compared to other state-of-the-art techniques, including PSO.

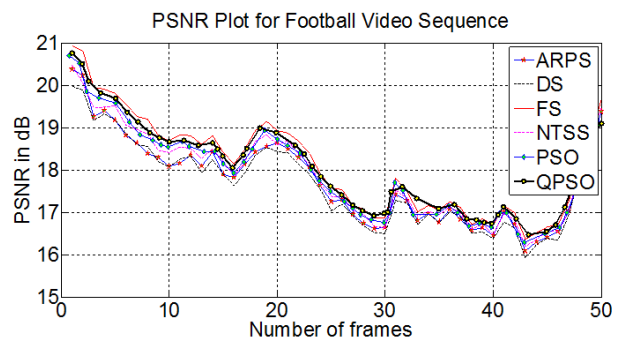


Fig. 11: PSNR plot for the Football sequence

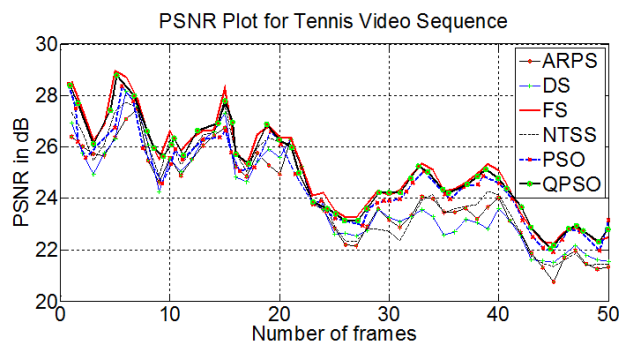


Fig. 12: PSNR plot for the Tennis sequence

Table 5: SSIM comparison of each video sequence.

Methods for ME	Garden	Football	Tennis	Clare
FS	0.8178	0.8835	0.811	0.7965
NTSS	0.7707	0.8606	0.6838	0.6875
DS	0.7664	0.8533	0.7195	0.7264
ARPS	0.802	0.865	0.7912	0.7632
PSO	0.8095	0.8793	0.7964	0.7813
Proposed	0.8102	0.8825	0.8034	0.7892

Fig. 9 shows a bar plot of the average PSNR for all four data sets. This plot shows that the average PSNR value for the proposed method was higher compared to the FS, DS, ARPS and the PSO based search methods. Table 5 shows the SSIM comparison, where the proposed method has good reconstruction compared to other methods. It is obvious that the SSIM index for FS will be higher. This is because all blocks are involved in the matching process as compared to the other techniques. However, the proposed method has a higher SSIM index than the other methods, except for the full search.

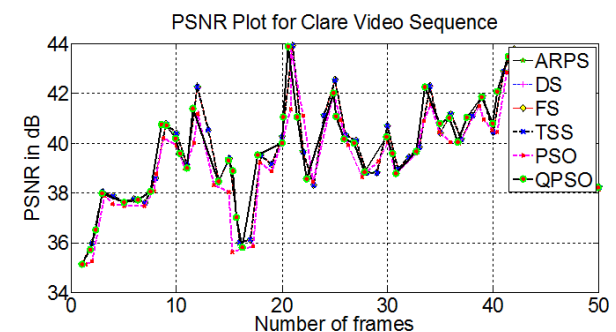


Fig. 13: PSNR plot for the Clare sequence

Table 6: Computation time (seconds) of various motion estimation algorithms.

Video Data set	FS	NTSS	DS	ARPS	PSO	Proposed
Football	3081.26	111.49	64.64	61.96	56.39	38.17
Tennis	3099.05	110.45	53.41	46.5	49.75	32.41
Garden	3062.48	106.27	49.09	40.9	44.87	28.76
Clare	3057.64	104.47	48.41	39.58	42.65	26.73

5.4 Complexity Analysis

The computation complexity depends on the time complexity and the number of operations being done. The computation time is illustrated in Table 6 and the computational complexity is given in Table 7. From Table 6, it can be seen that the proposed method takes less time for the computation of a motion vector for all four data sets. The complexity of the proposed method is only $1 + p \log p$ operations, which is very less than other state-of-the-art methods.

Table 7: Computational complexity of various search methods for motion estimation.

Search Method	Complexity
ES	$2*(2p+1)^2$
TSS	$[1+8\log_2(p+1)]$
NTSS	$[1+8\log_2(p+1)]+8$
4SS	$[18 \log_2(p+1)/4+9]$
ARPS	$2*(2*(p+1))$
PSO	p^2
Proposed	$1 + p \log p$

6. CONCLUSIONS

A novel pattern based motion estimation approach utilizing QPSO is proposed in this paper to provide a lower computational burden with enhanced accuracy. In this scheme, a pattern is predefined (HEXBS) initially followed by evolutionary based search techniques using the QPSO algorithm. QPSO is being used here to overcome the problem of entrapment at local optima, as well as yielding faster computation. The algorithm evades the problem faced by the conventional PSO. The validation of the proposed algorithm was done with various experiments. From the experimental results it can be concluded that the proposed approach outperformed the conventional methods in terms of PSNR, SSIM, SIR %, and the average number of search points. Various graphical representations were made for the justification of the proposed

method's superiority over other techniques.

This work can be further extended by going beyond this QPSO and using some other variants of PSO, namely CLPSO, DMS-PSO FIPS, ELPSO, CPPSO, and GOPSO, with an aim to achieve better block matching with lower computational complexity.

References

- [1] J. Cai and W. D. Pan, "On fast and accurate block based motion estimation algorithms using particle swarm optimization," *Information Sciences*, vol. 197, pp. 53-64, 2012.
- [2] Jalloul, M.K., and M. A. Al-Alaoui. "A novel Co-operative Motion Estimation Algorithm based on Particle Swarm Optimization and its multi-core implementation," *Signal Processing: Image Communication*, 39, pp. 121-140, 2015.
- [3] S. Immanuel Alex Pandian, G. Josemin Bala and J. Anitha, "A pattern based PSO approach for block matching in motion estimation," *Engineering Applications of Artificial Intelligence*, vol.26, Issue 8, pp. 1811-1817, 2013.
- [4] Z. Ping, W. Ping and Y. Hongyang, "A novel block matching algorithm based on particle swarm optimization with mutation operator and simplex method," *WSEAS Transactions on Systems and Control*, vol.6, Issue 6, pp.207-216, 2011.
- [5] 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. 438-442, Aug. 1994.
- [6] J. Lu and M. L. Liou, "A simple and efficient search algorithm for block-matching motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.7, no.2, pp. 429-433, Apr. 1997.
- [7] 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, Feb. 2000.
- [8] 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, Dec. 2002.
- [9] Y. Nie and K.K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *IEEE Transactions on Image Processing*, vol.11, no.12, pp. 1442-1449, 2002.
- [10] C. Zhu, X. Lin and L.P. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.12, no.5, pp. 349-355, May 2002.
- [11] E. Cuevas, D. Zaldivar, M. P. Cisneros, H. Sossa and V. Osuna, "Block matching algorithm for motion estimation based on Artificial Bee Colony (ABC)," *Applied Soft Computing*, vol.13, Issue 6, pp. 3047-3059, 2013.
- [12] Du Gy, Huang Ts, Song Lx, Zhao Bj, "A Novel Fast Motion Estimation Method Based on Particle Swarm Optimization," In proceedings of Fourth International Conference on Machine Learning and Cybernetics (August) , pp. 5038-5042, 2005.
- [13] Bakwad KM, Pattnaik S, et al. "Small Population Based Modified Parallel Particle Swarm Optimization for Motion Estimation," In proceedings of 16th International Conference on Advanced Computing and Communications, pp. 367-373, 2008.
- [14] Bakwad KM, Pattnaik SS, Sohi BS, Devi S, Golapudi SV, Sagar CV, Patra PK, "Fast motion estimation using small population-based modified parallel particle swarm optimisation," *International Journal of Parallel, Emergent and Distributed Systems*, 26(6) , pp. 457-476, 2011.
- [15] Huang YW, Chen CY, Tsai CH, Shen CF, Chen LG, "Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results," *Journal of VLSI signal processing systems for signal, image and video technology*, vol.42, Issue 3, pp. 297-320, 2006.
- [16] D. Tzovaras, I. Kompatsiaris and M. G. Strintzis, "3D object articulation and motion estimation in model based stereoscopic video-conference image sequence analysis and coding," *Signal Processing: Image Communication*, vol.14, Issue 10, pp. 817-840, 1999.
- [17] Barron JL, Fleet DJ, Beauchemin SS, "Performance of optical ow techniques," *International Journal of Computer Vision*, 12(1), pp. 43-77, 1994.
- [18] Skowronski J, "Pel recursive motion estimation and compensation in sub bands," *Signal Processing: Image Communication*, 14(5), pp. 389-396, 1999.
- [19] J. Jain and A. Jain, "Displacement Measurement and Its Application in Inter frame Image Coding," *IEEE transactions on Communications*, vol.29, no.12, pp. 1799-1808, Dec. 1981.
- [20] Zhou J A and Love, P E D and Wang, X and Teo, K L and Irani, Z, "A review of methods and Algorithms for optimizing construction scheduling," *Journal of the Operational Research Society*, 64, pp. 1091-1105, 2013.
- [21] Sun J, Lai CH, Wu XJ, "Particle swarm optimisation: classical and quantum perspectives," *CRC Press*, 2011.
- [22] Eberhart, Russell C and Shi, Yuhui, "Particle Swarm Optimization: Developments, Applications and Resources," *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE)*, pp. 81-86, 1995.
- [23] Sun, J., Xu W, & Feng, B. "A global search strategy of quantum-behaved particle swarm optimization," *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems*, vol. 1, pp. 111-116, 2004.
- [24] Sun J, Fang W, Wu XJ, Palade V, Xu WB, "Quantum-Behaved Particle Swarm Optimization: Analysis of Individual Particle Behaviour and Parameter Selection." *Evolutionary Computation*, 20(3), pp. 349-393, 2012.
- [25] Y. Ismail, J. B. McNeely, M. Shaaban, H. Mahmoud and M. A. Bayoumi, "Fast motion estimation system using dynamic models for H.264/AVC video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.22, no.1 , pp. 28-42, Jan. 2012.
- [26] Y.Cho CIPR Data set for YUV sequence <http://www.cipr.rpi.edu/resource/sequences/sif.html>, 2005.
- [27] Karthik, R., and R. Menaka, "Statistical characterization of ischemic stroke lesions from MRI using discrete wavelet transformations," *ECTI Transactions on Electrical Engineering, Electronics, and Communications*, 14.2, pp. 57-64, 2016.
- [28] Salhi Meriem, et al. "Mobility-Assisted and QoS-Aware Resource Allocation for Video Streaming over LTE Femtocell Networks," *ECTI Transactions on Electrical Engineering, Electronics, and Communications 13.1*, pp. 42-53, 2015.
- [29] Smita Pradhan and Dipti Patra, "RMI based non-rigid image registration using BF-QPSO optimization and P-spline," *AEU -International Journal of Electronics and Communications*, 69(3):609-621, 2015.



Yogananda Patnaik has received his M.tech. from Electrical Engineering Department, National Institute of Technology, Rourkela India in 2006. Currently he is pursuing Ph.D. at National Institute of Technology, Rourkela. His areas of interest are Signal / Image Processing / Video Processing, Computer vision.



Dipti Patra has received Ph.D. from National Institute of Technology, Rourkela India in 2006. Currently, she is working as Associate Professor in the Department of Electrical Engg., N.I.T., Rourkela, India. Her areas of interest are Signal / Image / Video Processing, Computer Vision and Visual Servo Control. She is a Senior Member of IEEE, Fellow of IETE and IETI, India.