

---

# Analysis of Particle Swarm Optimization in block matching algorithms for video coding

Kakalakannan Damodharan<sup>1</sup>, Thamarai Muthusamy<sup>2</sup>

<sup>1</sup>Department of Electronics and Communication Engineering, Theja Sakthi Institute of Technology for Women, Coimbatore, Tamilnadu, India

<sup>2</sup>Department of Electronics and Communication Engineering, Karpagam College of Engineering, Coimbatore, Tamilnadu, India

## Email address:

d.kamalkannan1@gmail.com (K. Damodharan), sreethamarai2000@yahoo.co.in (T. Muthusamy)

## To cite this article:

Kakalakannan Damodharan, Thamarai Muthusamy. Analysis of Particle Swarm Optimization in Block Matching Algorithms for Video Coding. *Science Journal of Circuits, Systems and Signal Processing*. Special Issue: Computational Intelligence in Digital Image Processing.

Vol. 3, No. 6-1, 2014, pp. 17-23. doi: 10.11648/j.cssp.s.2014030601.13

---

**Abstract:** Particle Swarm Optimization (PSO) is global optimization technique based on swarm intelligence. It simulates the behavior of bird flocking. It is widely accepted and focused by researchers due to its profound intelligence and simple algorithm structure. Currently PSO has been implemented in a wide range of research areas such as functional optimization, pattern recognition, neural network training and fuzzy system control etc.. In video processing PSO is used to find the best matching block in Block matching algorithm, bit rate optimization for MPEG 1/2, object tracking and data clustering. In this paper the usage of PSO in Block matching algorithms for video compression is analyzed and the results are compared with the existing techniques.

**Keywords:** Motion Vector, Sum Absolute Difference, Average Mean Square Error, Full Search Algorithms, Particle Swarm Optimization, Directed Particle swarm Optimization, Block Matching Algorithm, Memetic PSO, Mutation Simplex PSO

---

## 1 Introduction

PSO was introduced by [1],[2] Eberhart and Kennedy in 1995. It is a gradientless global optimization technique, suitable for continuous variable problems. This algorithm maintains a population of particles, where each particle represents the potential solution in the search space (Optimization). PSO has the following features:

- No evolutionary operators such as crossover and mutation as in GA.
- Ease of use with fewer parameters to adjust.

The aim of PSO is to find the particle position that results in the best evaluation of a given fitness function. Each particle fly through the N dimensional search space to find the optimum solution followed by the current better performing particle.

Each particle remembers its own best position  $X_{pbest}$  (that is, where the function was the fittest), and of all these ,the globally best value  $X_{gbest}$  is determined from the particles  $X_{pbest}$  values. As showed in expression (1), the particles are attracted by  $X_{pbest}$  and  $X_{gbest}$ .

At each iteration the particle velocity vector  $V$  and position

vector  $X$  are modified according to the equations (1) and (2).

$$V_{id}(t+1) = w \times v_{id}(t) + c_1 \times \text{rand}_1(.) \times (p_{id} - x_{id}) + c_2 \times \text{rand}_2(.) \times (p_{gd} - x_{id}) \quad (1)$$

$$X_{id}(t+1) = x_{id}(t) + V_{id}(t+1), \quad 1 \leq i \leq N, \quad 1 \leq d \leq D \quad (2)$$

where,  $N$  is the number of particles and  $D$  is the dimensionality;

$V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ ,  $v_{id} \in [-v_{max}, v_{max}]$  is the velocity vector of particle  $i$  which decides the particle's displacement in each iteration.

Similarly,  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ ,  $x_{id} \in [-x_{max}, x_{max}]$  is the position vector of particle  $i$  which is a potential solution in the solution space.

' $w$ ' is the inertia weight which decreases linearly during a run;  $c_1, c_2$  are both positive constants, called the acceleration factors which are generally set to 2.0;  $\text{rand}_1(.)$  and  $\text{rand}_2(.)$  are two independent random number distributed uniformly over the range  $[0, 1]$  and  $p_g, p_i$  are the best solutions discovered so far by the group and itself respectively. The velocity is calculated based on the contributions of the following three factors:

- Fraction of the previous velocity
- The cognitive component: The cognitive component is a function of the distance of the particle from its personal best position.
- The social component: The social component is a function of the distance from the particle found so far.

The personal best position of particle ‘i’ is calculated as in equation(3)

$$\begin{aligned}
 y_i(t+1) &= y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\
 &= x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t))
 \end{aligned}
 \tag{3}$$

where  $f(x_i(t+1))$  is the fitness value of  $x_i(t+1)$  and  $y_i(t)$  is the current best personal position of the  $i^{\text{th}}$  particle. The quality of the solution is measured by a fitness function.

Video compression plays an important role in video signal processing, transmission and storage. Since the available bandwidth for transmission is very limited, multimedia applications such as video conferencing, video on demand, video telephony and remote sensing are not possible without compression. Video is a sequence of correlated images. This temporal redundancy can be exploited for coding and processing. Motion estimation and compensation have been the most widely used methods in video compression and become the standard approach to reduce the temporal redundancies between frames. The block matching algorithm for motion estimation has been adopted in many international standards for digital video compression, such as H.264 and MPEG 4.

There are lot of fast searching techniques have been developed to search the best matching block in Block Matching Algorithm(BMA), Three Step Search, New Three Step Search, Diamond Search[3],and Adaptive Root Pattern Search[4] and Full Search Technique[FSA] are some searching techniques. But These techniques are based on the assumption that there are only one minima point in the error surface. So they stick the local minima problem. To overcome this GA based Block matching algorithm was developed [5],[6],[7]. The performance is similar to FSA and solves the local minima sticking problem at the cost of computational complexity [more than one quarter of the FSA].Therefore PSO is proposed for searching the matching block in Block Matching Algorithm. It leads to the development of a fast block matching algorithm for video motion estimation. Its performance is similar to GA, while the computational complexity is less. GA and PSO analyze the error surface in the search space as a multimodal function and hence the local sticking problem is eliminated in the both cases. When compared to GA, PSO has very few parameters to adjust. Parameters are insensitive to scaling of design variables, also Parameters can be parallelized for concurrent processing. Its computation complexity is less and easy for hardware implementation.

In addition an improved PSO search called Directed Particle Swarm Optimization [13] search is also used for motion estimation in which the best particles are searched with the predefined directions of the particles.

In this paper, PSO based Fast motion estimation algorithms

in Block matching for video coding concepts are discussed and their performances are analyzed. Section 2 explains the basic PSO based block matching, section 3 covers the bidirectional frame prediction using PSO, and in section 4 Directed particle swarm optimization method is considered. Section 5 explains the recently developed PSO based Block matching algorithms such as Memetic PSO and Mutation Simplex PSO. The experimental results and conclusion are presented in section 6 and section 7 respectively.

## 2. Block Matching Algorithm Using PSO

Motion estimation is a process which determines the motion between two or more frames. Motion compensation uses the motion information and a given reference frame to reconstruct video frames. Because of intensive computations and the large amount of resources required by motion estimation, it has been an active research field in the past decades and various algorithms have been developed. Among them one group of methods that are most widely used are block based techniques called Block Matching Algorithms (BMAs).

Till now a lot of research work has been done on developing fast efficient block matching algorithms. They aim at reducing the computation of motion estimation as much as possible while maintaining compensated frame quality as good as possible, thus improving the performance of the video codec as a whole.

In BMA techniques a frame is divided into non overlapping blocks of size 8x8 or 16x16 and the pixels on each block are assumed to have uniform motion. Each block is compared with candidate blocks in the reference frame within the search area to obtain the motion vector (mv). The motion vector is computed by subtracting the co ordinates of the macro block in the reference frame from that of the matching block in the target frame. Figure 1 illustrates the motion estimation.

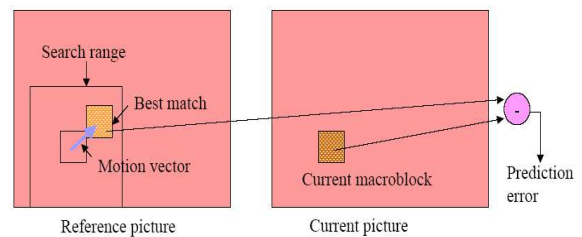


Figure 1. Block Matching algorithm –motion estimation.

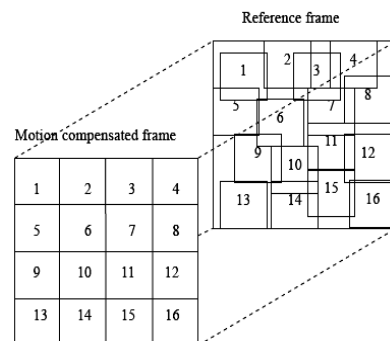


Figure 2. An illustration of motion Compensated frame.

As shown in Figure 2, the current frame is divided into 16 macro blocks of size 8x8. For each macro block the best matching block is searched in the reference frame and is shown with block number.

**2.1. PSO Based Block Matching**

Particle Swarm Optimization technique [8] is used to find the best matching block for the block in the current frame. In the optimization point of view, block based methods are considered as minimization problem. The matching error between the current block and its best matching block in the reference frame should be minimum. The matching error is positive when the motion of the block is towards positive direction from K-1th frame to Kth frame and negative when the motion of the block is in negative direction from K-1th frame to Kth frame. The goal is to find the best displacement motion vector for each macro block in the sense of satisfying the criteria.

Sum Absolute Difference equation:

$$SAD = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N (|I_k(i, j) - I_{k-1}(i, j)|) \quad (4)$$

Fitness equation and performance evaluation average mean square error  $\sigma_\epsilon$  is calculated as per equation (5)

$$\sigma_\epsilon = \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^N ((I_{org}(i, j, k) - I_{com}(i, j, k))^2) \quad (5)$$

$$PSNR = 10 \log \left( \frac{I_{max}^2}{\sigma_\epsilon} \right) \quad (6)$$

Where K is the number of frames in the video sequence.

**2.2. Basic Block Matching Algorithm Using PSO**

- Randomly initialize the particle in eight directions with random initial velocity in the macro block.
- For each particle P evaluate the sum absolute difference as in equation above .(4)
- Update the personal best and global best based on fitness value in eq (3)
- Update the position and velocity as in eqs (1 )& (2)
- Repeat until the stopping criterion is met.

**2.3. Stopping Criteria**

Two methods are followed to stop the search procedure in PSO.

One is by fixing the maximum number of iterations. Once the maximum iterations reached the global best is taken as the optimum solution. Another one is by fixing very small threshold value. Here each and every iteration the variation of  $P_g$  is monitored. Over a period of time (iterations), if the change in  $P_{g_{best}}$  is less than the specified threshold then the algorithm is stopped. In the block matching search problem, fixed iteration method is followed due to the centre biased characteristics of the real world motion fields.

In video sequence moving objects and background appears

still in successive frames. So motion vector for this still background or object is zero. The computations performed to find the motion vector in this case is waste of time. In order to minimize the computation complexity, Madhan Mohan Manohar introduced new algorithm with PSO BMA, with zero motion prejudgment(ZMP)[9]. He fixed some minimum threshold value based on the nature of the video sequence.

First the SAD is calculated and if SAD is less than the threshold, the block is considered as static and its motion vector is written as zero. In order to reduce the computation complexity and improve the PSNR value particle's initial positions are arranged as shown in Figure.3 one of the four patterns below for different regions of the frame.

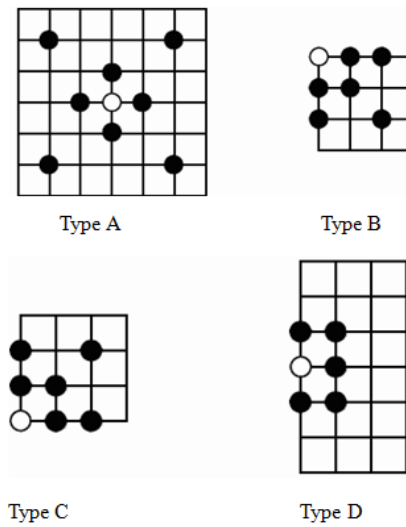


Figure 3. Patterns of initial particles

Top leftmost corner macro block, the particles are arranged as in pattern B and PSO routine is performed. Pattern D – for leftmost column MBs ,Pattern C- for bottom left corner and pattern A is commonly used for all other macro blocks.

**3. Bidirectional Frame Prediction Using PSO**

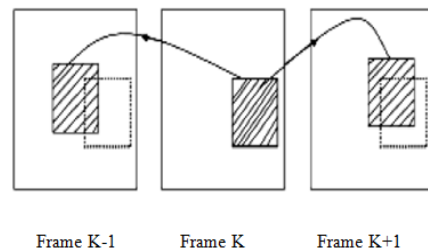


Figure 4. Bi directional motion estimation

An efficient bidirectional motion estimation technique, which is based on PSO, was introduced by D.Ranganatham et.al [10 ].In [10 ] the best matching block is found by means of taking two reference frames(previous frame and future frame). This algorithm is developed for MPEG 4 codec, which uses three different types of frames(I Frame-Independent

frame, P frame-predictively coded frame and B Frame- Bi directionally predictively coded frame). To predict the B-frame bi directional motion estimation is used as shown in figure. A linear combination of forward and backward motion information are used for the bi directional frame prediction

If we apply motion estimation using PSO in both directions separately, it is time consuming process and it needs some measures to find the best mv among the two values. So bi directional block matching algorithm using PSO becomes necessary. The algorithm is as follows

- Static macro block prediction
- Particle initial position as in format A(as in fig 3)
- Each time find the matching error (SAD) point in the past frame (SAD<sub>P</sub>) and the future frame (SAD<sub>F</sub>) as shown in Figure 4.
- Take the minimum out of both matching error based on the cost function
- Cost Function = min (SAD<sub>P</sub>,SAD<sub>F</sub>) Until the stopping criteria it will continue the above steps. Fixed number of iteration stopping criteria is adapted.
- Save the final motion vector for motion compensation.

#### 4. Directed Particle Swarm Optimization

A number of works have been done on the analysis and development of PSO since its introduction.

In conventional PSO algorithm, the search velocity  $v(n)$  is clamped within a range, denoted by  $V_{max}$ . Given an optimization problem, the proper range of  $V_{max}$  for good performance is always limited and hard to be predicted. Hence, a PSO with decreasing  $V_{max}$  method (PSO-DVM) is developed [11], in which  $V_{max}$  is decreasing over time. By using this method, a large scale of searching is expected at the early steps, so that the population can remain in enough diversity profitable to converge to the global optimum. As the searching process continues, the searching scale is reduced to allow the solution to be found.

Another improved paradigm called guaranteed convergence PSO (GCP SO) [12], is developed specifically to address the drawback of the standard PSO. When the position of a particle equals its personal best position or the global best particle, the velocity is influenced by the inertial term. Therefore, if this particle stays on the global best position, which is also the personal best position, its velocity tends to be zero and the particle stagnates over a number of iterations.

To solve this stagnation problem, the velocity update for the global best particle is changed, while the other particles remain as the original updating principle. In some degrees, it is viewed as a mutation method, where the mutation behavior is selective. This means only the particle in stagnation state is relocated randomly. A similar operation is adopted in the PSO with a “mutation” and time-varying acceleration coefficient (MPSO-TVAC) paradigm.

All these four improvements on PSO have the same character that all of them use the basic updating principles with inertia weight or constriction coefficient to adjust particles’ positions.

In the proposed Directed PSO [13], the new positions are calculated by performing single-point crossover operation with the existing position as performed in GA. This operation avoids the local minima and leads to find the optimum blocks with minimum computational complexity.

And a new parameter called Velocity Rate ( $V_R$ ) is introduced to control the updation of velocity based on the performance history of the particles. Initially all particles are assigned ‘1’ as velocity rate. At each iteration, based on the fitness value the  $V_R$  is either increased or decreased by 0.1 for each particle. If  $V_R$  value of the particle which gives the optimum value is increased 0.8, and the updated velocity is multiplied with  $V_R$ , the equation (1) is modified as

$$V_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand_1(.) \times (p_{id} - x_{id}) + c_2 \times rand_2(.) \times (p_{gd} - x_{id}) + V_R \tag{7}$$

One more parameter is introduced in that modified PSO (DPSO), which is the direction (angle) of the particles. Here we have eight different directions,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ,  $180^\circ$ ,  $225^\circ$ ,  $270^\circ$ , and  $315^\circ$  and from these the particles can choose any one direction at random to select the optimum value, but the condition is that all the particles have to move in the same direction. With these parameters, the PSO can avoid premature convergence.

##### 4.1. Block Matching Using DPSO and Clonal PSO

Based on the center-biased characteristics in video sequences, the encoder creates a prediction of a region of the current frame based on previously encoded and transmitted frames. If the frame is processed in raster order, the current-encoded MB should have four patterns of region of support (ROS) that consists of the neighboring blocks whose MVs will be used to compute the predicted MV for prediction due to the limited computational cost. The ROS pattern D is adopted and motion blocks are estimated as discussed in [10].The algorithm is described in paper[13]

Clonal PSO (CPSO) block matching algorithm is proposed by Gorpuni [14]. This algorithm employs Clonal mechanism found in the natural immune system of creatures into PSO by cloning the best individual of every ten succeeding generations, CPSO has a better optimization solving capability and convergence performance than the conventional PSO and GA.

#### 5. Block Matching Using Memetic PSO and Mutation Simplex PSO

Memetic PSO is also used for motion estimation in video coders [15]. Memetic algorithm incorporates local search technique in the standard PSO algorithm for effective optimization.

The position equation of the standard Particle Swarm Optimization is modified and adaptive step size is achieved by using time varying inertia weight. The performance of the Memetic algorithm was good as compared to existing algorithms in terms of Number of Computations and accuracy.

To get the step size, the velocity and position equations of PSO are modified as given below. The velocity equation is expressed as Equation(8) in [15].

$$V(t) = w * C * r \tag{8}$$

Where  $w$  is the inertia weight,  $C$  is the acceleration constant,  $r$  is random number between 0 to 1 and  $t$  is generation number. To get the adaptive step size, the time varying inertia weight ( $w$ ) is used instead of constant inertia weight similar to GLBestPSO for getting the true motion vector dynamically. The time varying inertia weight is based up on previous motion vectors as given in Equation (9) in [15]

$$\begin{aligned} W &= (1.1 - G_{best} + P_{best}) \\ G_{best} &= X + Y \\ P_{best} &= X - Y \end{aligned} \tag{9}$$

Where,  $X$  and  $Y$  is the x and y coordinates of the predicted motion vector. The velocity term in Equation (8) is added with previous motion vector to predict the next best matching block (position  $S(t+1)$ ) as given in Equation (10). (Where  $S(t)$  position of the current best Matching Block)

$$S(t+1) = S(t) + v(t) \tag{10}$$

In Memetic Algorithm, a search is made in an earlier frame of the sequence over a random area of the frame. The search is for the best matching block viz. the position that minimizes a distortion measured between the two sets of pixels comprising the blocks. The relative displacement between the two blocks is taken to be the motion vector. Usually the macro block is taken as a square of side consists of 16 pixels. The compression ration is 128:1 or 256:2. The each block size of 16 x 16 is compressed into two pixels which are nothing but motion vectors. Here Mean Absolute Difference (MAD) is used as a fitness function.

Zhang Ping et al [16] proposed a Novel Search Algorithm based on Particle Swarm Optimization and Simplex search Method for Block Motion Establishment. This algorithm utilizes the global search ability of PSO and the local search ability of simplex method in order to improve the performance of motion estimation in video coding. In order to accelerate the convergence of PSO and improve the precision of local search, mutation operator and simplex method are used.

Simplex method (SM) is a multi-dimensional unconstrained optimization method that converges rapidly towards minimum point in small or irregular search areas. A D-dimensional simplex is a geometrical figure which consists of (D+1) points. A non-degenerate simplex is one that encloses a finite inner D-dimensional volume. The SM method starts with (D+1) initial points randomly in the search space and then calculates the fitness value of each point.

In the two-dimensional simplex search, a search triangle is used to locate a minimum of the performance index or error function. This error function is evaluated at the triangle vertices which represent possible minimum locations. The locations of the triangle vertices are modified in a manner that

moves the triangle towards possible minimum locations by moving the triangle away from locations of high error function values.

It finds the best point  $X_b$  where fitness function is lowest, the worst point  $X_w$  where fitness function is highest and the second worst point  $X_n$  where fitness function is second higher. Then SM method takes a series of operations include reflection, expansion, and contraction to find a better point and replace the worst point. This process is repeated until a termination criterion is satisfied. Finally one optimization point is found.

The calculations of reflection, expansion, contraction are as follows as per equation (11) in [16]:

$$\begin{aligned} X_r &= X_c + \alpha(X_c - X_w) && \text{if } f_b < f_r < f_n \\ X_e &= X_c + \gamma(X_r - X_c) && \text{if } f_r < f_b \\ X_s &= X_c + \beta(X_w - X_c) && \text{if } f_n < f_r \end{aligned} \tag{11}$$

Where  $X_c$  is the centroid of remaining points,  $X_r$  is the reflection point,  $X_e$  is the expansion point,  $X_s$  is the negative and positive contraction point as shown in Figure.  $\alpha$ ,  $\gamma$  and  $\beta$  are coefficients of reflection, expansion, contraction,  $f_b, f_r, f_n, f_w$  are the values of fitness function on point  $X_b, X_r, X_n, X_w$  respectively.

In this method, first the conventional PSO search is performed. Next this simplex search is initiated to replace the three worst particles (Blocks) which are obtained during the PSO search.

## 6. Experimental Results

First the performance of the following methods

- (i) PSO-ZMP
- (ii) Bi directional frame prediction using PSO
- (iii) Directed Particle swarm optimization

are analyzed for the standard video sequences like foreman, akiyo, news and Claire.

The performance is compared with the standard motion estimation techniques of Diamond Search and Adaptive Root Pattern Search.

The performance is evaluated in terms of average mean square error, Average search points/Macro Block and peak signal to noise ratio per frame of the reconstructed video sequence.

The input video sequence at normal frame rate (30 frames per second) if decomposed into 8x8 square blocks. For each block, the best matching block in the reference frame is calculated. Before that the static macro block prediction procedure is done based on the threshold value as in Table-1

Table 1. Threshold for the four test video sequences

Sequence	Assumed threshold	
	Format	Threshold
Akiyo	QCIF	380
News	QCIF	250
Foreman	QCIF	270
Claire	QCIF	320

PSO parameters:

The parameters  $c_1$  &  $c_2$  are taken as 2.2 and the inertia weight  $w=0.9$ . It linearly decreases from iteration to iteration. The number of particles is 10. The particles are arranged as in Type A. The maximum  $X_{max}$  range value is set as 16 and the  $V_{max}$  equals 10. The Number of iterations set to 25. The average mean square error and PSNR values are given in Table 2 and Table 3 respectively. In the Directed PSO case the initial Velocity rate parameter  $V_R$  is taken as 1. The video frame rate is 30 Frames /sec.

All the PSO based methods the average mean square error is comparatively less with the standard techniques. For all the four video sequences the error in Directed PSO is less and its

computational complexity is also less.

The Bi directional PSO+ZMP, the current frame is compared with the previous and future frames. So the complexity is doubled. Similarly the PSNR values of all the video sequences are high in the case of Directed PSO. It is almost 1db greater than the ARPS, PSO+ZMP, BiPSO+ZMP foreman sequence, which contains rapid movements among the frames.

Table 4 describes the computational complexity of the above mentioned algorithms. It is given in terms of no. of additions, subtractions and multiplications are required to search the best matching block in the reference frame.

Table 2. Average mean square prediction error.

Sequence	Diamond search	ARPS	PSO+ZMP	Bi-PSO+ ZMP	Directed PSO
Akyio	17.87	14.25	14.16	15.67	14.14
News	87.67	79.45	76.34	75.67	76.45
Foreman	178.45	150.24	164.36	152.48	152.56
Claire	149.12	146.29	149.27	148.47	147.12

Table 3. PSNR values at normal frame rate

Sequence	Diamond search	ARPS	PSO+ZMP	Bi-PSO+ZMP	Directed PSO
Akyio	43.50	43.49	42.07	42.37	42.56
News	35.52	36.35	34.97	35.02	38.44
Foreman	35.45	35.83	34.64	35.82	36.12
Claire	36.34	36.82	35.78	36.97	38.23

Table 4. Computational gain for various techniques

Sequence	PSO to DS	DPSO to Bi PSO+ZMP	DPSO to DS
Akyio	5.3346	2.4586	6.6743
Foreman	4.6324	2.9837	5.3892

6.1. Comparison of Memetic PSO with MSPSO in Block Matching

The input video files are Foreman.avi and Rhino.avi. The four step search method is selected to compare with MSPSO and Memetic algorithm. The input parameters of simulation of MSPSO are as follows  $c_1=c_2=2$ ,  $K=0.729$ ,  $\alpha=1$ ,  $\beta=0.5$  and  $\gamma=2$

In the simulations, the two test sequences Foreman.avi and Rhinos.avi are taken. The reference frame of motion estimation is the previous frame, that is current frame is estimated by previous frame. The fixed size of macro block is 16\*16 and the search range is (-16, + 16).

Mean absolute difference is used as a fitness function in Memetic PSO and Sum Absolute difference is used as a fitness function in MSPSO technique.

Table 5. Comparisons of PSNR of 4SS, MSPSO and Memetic for Foreman.avi

Algorithms	PSNR (dB)				
	Frame 1	Frame 2	Frame3	Frame4	Frame5
4SS	32.15	31.36	31.18	31.26	31.39
MSPSO	43.23	43.33	43.68	43.55	43.99
Memetic	32.96	32.34	31.24	31.78	31.82

Table 6. Comparisons of PSNR of 4SS, MSPSO and Memetic for Rhino.avi

Algorithms	PSNR(dB)				
	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5
4SS	35.03	37.39	35.47	35.14	34.79
MSPSO	53.06	52.92	50.41	50.01	51.03
Memetic	34.64	38.38	34.14	34.5	33.71

Table 7. Comparisons of no. of computations of 4SS, MSPSO and Memetic for Foreman.avi

Algorithms	Number of Computations				
	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5
4SS	24.07	24.19	23.74	23.99	23.88
MSPSO	9.08	7.02	10.63	7.07	7.89
Memetic	15.96	15.98	15.96	15.96	15.96

Table 8. Comparisons of no. of computations of 4SS, MSPSO and Memetic for Rhinos.avi

Algorithm	No. of Computations				
	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5
4SS	24.31	24.02	24.2	24.09	24.38
MSPSO	12.23	12.03	12.34	12.66	12.89
Memetic	16.96	16.34	16.46	16.4	16.22



The results of Table 5 and Table 6 show the PSNR values of the first five frames in the Foreman sequence and Rhinos Sequence respectively. The results denote the difference between PSNR of Four step search, MSPSO and Memetic. MSPSO which is a combination of global optimization ability of PSO with mutation and local optimization ability of SM, makes the particle to overcome from its local convergence and provides global best optimum solution. The simulation result shows that MSPSO gives the better PSNR value than other two algorithms

The computational complexity of the Four step search, MSPSO, Memetic algorithms are obtained by the number of searching points in the motion estimation of the test video sequences. The results of Table 7 and Table 8 denote that the difference between computational complexity of the Four step search, MSPSO and Memetic. The MSPSO method uses the early termination of search. So the simulation result shows that MSPSO method has less computational complexity than the other two algorithms.

## 7. Conclusion

In this paper various types of PSO based motion estimation for video compression are analyzed with the standard techniques such as Diamond search, ARPS and 4SS. The PSO search technique eliminates the local minima sticking problem which appears in traditional fast searching techniques and its computation complexity is less when compared with all other techniques. At the same time its performance is similar to that of the Full Search Techniques. In directed PSO, the directional properties of the video sequences are utilized, which minimizes the computational complexity. The performance of the PSO based block matching algorithms are improved by making modifications in its velocity equation and adding local search techniques such as Simplex Method along with the PSO Search. In future the multi objective PSO may be used to improve the performance of the PSO based block matching techniques for video compression.

## References

- [1] J. Kennedy, R.C Eberhart, "Particle swarm optimization", in IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.
- [2] J. Kennedy, R.C Eberhart, "A discrete binary version of the particle swarm optimization algorithm", in IEEE International Conference on Neural Networks, Perth, Australia, pp. 4104-4108, 1997.
- [3] Y.Nie and K.K.Ma, "Adaptive rood pattern search for fast block matching motion estimation", IEEE Trans. Image Processing vol.11, No.12, PP.1442-1449, Dec. 2002.
- [4] Shan Zhu and Kai-Kuang Ma, "A New Diamond search algorithm for Fast Block Matching Motion Estimation", IEEE Trans. Image Processing. Vol9 no.2 pp.287-290, February 2000.
- [5] C.H. Hsieh, P.C. Lu, J.S. Shyn and E.H. Lu, "Motion estimation algorithm using interlock correlation", IEEE Electronic Letters, vol. 5, pp. 276-277, 1990.
- [6] M. Ghanbari, "The cross-search algorithm for motion estimation", IEEE Transaction on Communication, pp. 950-953, 1990.
- [7] K. Chow and M. Liou, "Genetic Motion search algorithm for video compression", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, pp. 440-445, 1993.
- [8] Xuedong Yuan, Xiaoqing Shen "Block Matching Algorithm Based on Particle Swarm Optimization for motion estimation", The 2008 International conference on embedded Software and Systems (ICCESS2008), PP 191-197.
- [9] R. Ren, M. Manokar, Y. Shi, B. Zheng, "A Fast Block Matching Algorithm for Video Motion Estimation Based on Particle Swarm Optimization and Motion Prejudgment", Proc. IEEE International Conference on Industrial and Information Systems. 2007.
- [10] D. Ranganadham, and P. Gorpuni, "An efficient bidirectional frame prediction using particle swarm optimization technique", International Conference on Advances in Recent Technologies in Communication and Computing, 5328092, pp. 42-46, 2009.
- [11] H. Y. Fan and Y. Shi, "Study on Vmax of particle swarm optimization," in Proc. Workshop Particle Swarm Opt., Indianapolis, IN, 2001.
- [12] F. van den Bergh, "An analysis of particle swarm optimizers," Ph.D. dissertation, Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, 2002.
- [13] Thamarai M and Shanmugalakshmi R "Video coding using Directed Particle Swarm Optimization' CiiT International Journal of Digital Image Processing, vol. 2, no. 8. pp. 2010.
- [14] Gorpuni PK, 'Development of Fast motion Estimation Algorithms for Video compression', M. Tech Thesis report. National Institute of Technology, Rourkela, 2009.
- [15] Pooja Nakpal & Baghla S, 'Video Compression by Memetic Algorithm', International Journal of Advanced Computer Science Applications, vol. 2, no. 6, pp. 142-145, 2011
- [16] Zhang Ping, Wei Ping, Yu Hong-yang & Fei Chun, 'A Novel Search Algorithm Based on Particle Swarm Optimization and Simplex Method for Block Motion Estimation', International Journal of Digital Content Technology and its Applications, vol. 5, no. 1, pp. 76-86, 2011