A Novel Block Matching Algorithm Based on Particle Swarm Optimization with Mutation Operator and Simplex Method

ZHANG PING, WEI PING, YU HONGYANG
School of Electronic Engineer
University of Electronic Science and Technology of China
Jianshe North Road Sec.2 No.4 (610054)
CHENG DU, CHINA
pingzh@uestc.edu.cn, weipin@ee.uestc.edu.cn, hyyu@ee.uestc.edu.cn

Abstract: -To improve the performance of motion estimation in video coding, we propose a novel block matching algorithm, which utilize the global search ability of particle swarm optimization (PSO) with mutation operator and the local search ability of simplex method (SM).According to the center-biased and temporal-spatial correlation feature of motion vector and the global randomness of PSO, the fixed and random points are selected as the initial individual. Then, block matching process is executed by the updating of the position and velocity of individual. In order to accelerate the convergence of PSO and improve the accuracy of local search, mutation operator and simplex method are used. Meanwhile, based on the feature of static macro blocks, the proposed algorithm intelligently uses early termination strategies. Experimental results demonstrate that the proposed algorithm has better PSNR values than conventional fast block matching algorithms especially for video sequences with violent motion while the computational complexity of the proposed algorithm has negligible increase.

Key-Words: - motion estimation, block matching algorithm, particle swarm optimization, mutation, simplex method, early termination

1 Introduction

Motion estimation (ME) is designed to reduce the temporal redundancy between frames of video sequences. Since approximately 60-80% of total computation time of video codec is expended by motion estimation, the motion estimation algorithm has a significant effect on the performance of the entire codec. Due to their simplicity and reasonable performance, the block matching algorithms (BMAs) are the most popular algorithms for motion estimation and widely used in various video coding standards such as MPEG-X and H.26X series [1] [2]. In BMAs, a video current frame is partitioned into non-overlapping blocks of equal size. And the best matched block is determined using certain matching criteria within a predefined search window.

Among the various BMAs, the full search (FS) algorithm is the most optimal, because it can find the best matching block in the search area. However, it requires enormous computations due to the large number of searching and comparing for all possible candidate blocks. Therefore, many classical fast algorithms were proposed to reduce the computational complexity. The most popular algorithms were conventional methods such as four step search (FSS) [3], diamond search (DS) [4] [5], adaptive rood pattern search (ARPS) [6]. Base on these conventional methods, many improvement algorithms were proposed, such as some algorithms [7] based on temporal-spatial correlation feature and some algorithms [8] based on adaptive mode switching and so on. These fast algorithms are easy to be realized and can reduce the number of search points. However, these algorithms are base on unimodal error surface assumption, which is not always validity in real-world video sequences. Especially for sequences with violent motion, local minimum points can spread over the search window, thus these fast algorithms are likely to fall into local minima and cannot acquire accurate motion estimation.

Recently, probabilistic searching method such as particle swarm optimization (PSO) [9][10][11][12] is successfully applied to motion estimation problems. PSO algorithm utilizes global optimization characteristics of swarm intelligent to acquire better global estimation, especially for video sequences with violent motion. However, the computational complexity of PSO algorithm is very high because

the global optimal solution is finally found after finishing several iterations. Meanwhile, PSO algorithm easily falls into prematurity and local optimization, thus the search accuracy of PSO algorithm is limited.

For these reasons, this paper proposes a novel block matching algorithm (MSPSO), based on PSO with mutation operator and simplex method (SM) [18] [19]. Because simplex method is a multi-dimensional unconstrained optimization method that converges rapidly towards minimum point in small or irregular search areas, thus MSPSO algorithm combines the powerful global search ability of PSO and the high accurate local search ability of SM, overcomes the disadvantage of previous block matching algorithms of PSO. Furthermore mutation operation is added to avoid premature convergence, which expands the search area to find the best matching block. The experimental results demonstrate that the MSPSO algorithm has better PSNR values than conventional fast block matching algorithms especially for video sequences with violent motion while the computational complexity of the proposed algorithm has negligible increase.

This paper is organized as follows: Section II describes the particle swarm optimization and simplex method. The fast block matching algorithm (MSPSO) is given in section III. In section IV the simulation results are given and the performance of the algorithms is compared. Section V concludes the whole paper.

2 Particle Swarm Optimization and Simplex Method

2.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is a search optimization technique which has the characteristics of both evolutionary computation and swarm intelligence. It was first introduced by Kennedy and Eberhart in 1995 [13], which mimics swarm behavior in birds flocking and fish schooling [14]. The basic idea of PSO is to find the best optimal solution through collaboration and information sharing among individuals in the swarm. In PSO, each particle flies in the search space with a velocity which is dynamically adjusted according to its own flying experience and its companions' flying experiences. As PSO is easy to implement, it has rapidly progressed in recent years and with many successful applications seen in solving real-world optimization problems [15] [16] [17].

In PSO, each particle i is associated with two vectors, i.e., the velocity vector $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$ and the position vector $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$, where D stands for the dimensions of the solution space. The velocity and the position of each particle are initialized by random vectors within the corresponding ranges. During the evolutionary process, the velocity and position of particle i on dimension d are updated according to the following equations:

$$v_{id} = v_{id} + c_1 rand_{1d}(pBest_{id} - x_{id}) + c_2 rand_{2d}(gBest_d - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

where c_1 and c_2 are the acceleration coefficients, which control how far a particle will move in a single iteration. $rand_{1d}$ and $rand_{2d}$ are two uniformly distributed random numbers in dependently generated within [0,1] for the d th dimension. In (1), $pBest_{id}$ is the position with the best fitness found so far for each particle, and $gBest_d$ is the best position in the neighborhood.

A user-specified parameter $V_{\max d} \in \Re^+$ is applied to clamp the maximum velocity of each particle on the d th dimension. Thus, if the magnitude of the updated velocity $|V_{id}|$ exceeds $V_{\max d}$, then V_{id} is assigned the value $sign(V_{id})V_{\max d}$. In this paper, the maximum velocity $V_{\max d}$ is set to 20% of the search range.

The constriction factor has been introduced into PSO for analyzing the convergence behavior, i.e., by modifying (1) to,

$$v_{id} = \kappa[v_{id} + c_1 rand_{1d}(pBest_{id} - x_{id}) + c_2 rand_{2d}(gBest_d - x_{id})] \quad (3)$$

where the constriction factor κ

$$\kappa = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \quad (4)$$

κ is set to 0.729 with $\varphi = c_1 + c_2 = 4.1$. Where c_1 and c_2 are both set to 2.05. The constriction factor

and the acceleration coefficients are important parameters in PSO.

2.2 Simplex Method

Basic simplex method was presented by Spendleye et al, which does not use a local model of object function and works without the assumption of continuity. Then it was improved by Nelder and Mead, to what is called the nonlinear simplex method. Simplex method was proved to be widely used in many areas of economics and engineering, and it also was successfully applied to motion estimation problems [20] [21].

Simplex method 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:

$$\begin{aligned} X_r &= X_c + \alpha(X_c - X_w) \qquad if\ f_b < f_r < f_n \\ X_e &= X_c + \gamma(X_r - X_c) \qquad if\ f_r < f_b \\ X_s &= X_c + \beta(X_w - X_c) \qquad if\ f_n < f_r \end{aligned} \tag{5}$$

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 Fig.1. α , γ and β 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.

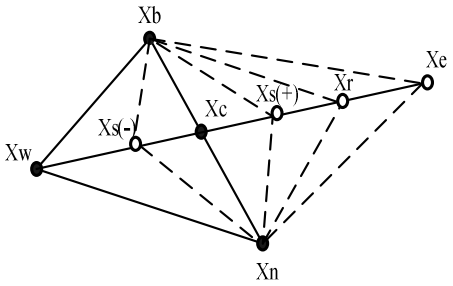


Fig.1 Different points in simplex search

3 Particle Swarm Optimization with Mutation and Simplex Method (MSPSO) Algorithm for Block Matching

In block matching algorithms of motion estimation, one frame is divided into non-overlapping macro blocks, with the size of 16*16 typically. Each block of current frame is compared with candidate blocks in the search area to find out the best matching candidate block.

In MSPSO algorithm, both PSO and SM are applied to motion estimation, best matching block of motion vector is found by utilizing the global search ability of PSO and the accurate local search ability of SM. Furthermore mutation operation is added to avoid premature convergence, which expands the search area to find the best matching block.

The multiple motion vectors of candidate macro blocks are represented as “particle” of PSO. And the horizontal and vertical coordinates of motion vector are represented as two positions of the particle. When the appropriate initial habitat has been selected, the best matching macro block can be found by the iterative calculation, the constant comparison and the updating of particles’ position. In order to improve the search efficiency of algorithm, the following MSPSO algorithm is discussed as follows:

3.1 The definition of fitness function

The fitness function of PSO is defined as SAD between the pixel values in current block of current frame and the block with corresponding location of the reference frame.

$$SAD(i,j)=\sum_{i=1}^{N}\sum_{j=1}^{N}\left|f_k(x,y)-f_{k-1}(x+i,y+j)\right| \quad (6)$$

Where, $f_k(x,y)$ is pixel value of pixel point (x,y) in the k-th frame, N is the size of macro block in motion estimation, i and j are horizontal and vertical position of macro block.

3.2 Improved initial population

This paper selects certain fixed points and random points as the initial population. This method utilizes temporal-spatial correlation of the motion vector and maintains the random characteristics of PSO.

According to the analysis of motion vector, motion vector distribution has both the center-biased feature and temporal-spatial correlation feature. As shown in Fig.2, current macro block #0 has similar motion vector with three adjacent macro blocks (left #1, upper #2 and upper-right #3) and the macro block #4 of the same location in the reference frame.

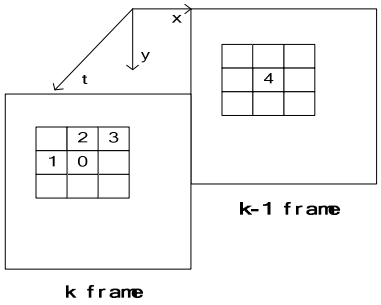


Fig.2 Relative position of current macro block and other macro blocks

Therefore the points of block #1, #2, #3, #4 are selected as the choices of fixed initial population points.

At the same time, according to formulation (7), generate some random points as the other part of initial population. The number of random points is determined according to population size.

$$P_{i,j}=\lfloor x_{\min}+(x_{\max}-x_{\min}+1)*rand(0,1)\rfloor \quad (7)$$

$x_{\min}$ and $x_{\max}$ are the search boundary and $rand(0,1)$ is a random number uniformly distributed of (0, 1).

3.3 Partial particle mutation

Standard PSO algorithm relies on the cooperation and competition between particles. However, particle does not have a mutation mechanism, once one particle once is constrained by a local best, it is difficult to jump out of the local best's restraint and need help from other particles' successful discoveries. Therefore, during each iterating, not only particle's $pBest_{id}$ and $gBest_d$ should be updated, but also the particles which have the worst fitness value should be eliminated.

In this paper, once K particles which have the worst SAD value are eliminated, K new particles are re-created within the search domain by using mutation operator.

This paper use Guass distribution as mutation operator. Thus, K new particles are re-created as follow:

$$Pnew_{i,j}=\lfloor x_{\min}+(x_{\max}-x_{\min}+1)*N(0,1)\rfloor \quad (8)$$

Where $N(0,1)$ is a random number which obeys standard Guass distribution.

3.4 Early termination of search

In video sequence, most blocks are stationary or quasi-stationary, which are called as static macro blocks. The SAD values of static macro blocks are very slow. Once the standard PSO algorithm is used for the searching of the static blocks, the calculation work will increase on the contrary. The process of searching should be terminated when the current macro block is a static block. During the searching process, once the SAD value is less than the threshold value, the current block should be determined to be the static block and the searching should be stopped. Otherwise it should be determined to be non-static block and the searching continues.

3.4 SM searching

Probabilistic searching methods including PSO are dedicated to find global optimization, however their abilities for local optimization are weak. Therefore, after finishing each iterating of PSO in MSPSO algorithm, SM searching is implemented to find three better points to replace three worst points.

In motion estimation, simplex method is a two-dimensional optimization method, a simplex is a triangle. Thus three worst points, which are formed at the end of each iterating of PSO, are chosen to define the initial vertices. The fitness function of vertices is also SAD. The coefficients values of reflection, expansion and contraction are α, γ and β.

The detail of SM searching is as follows:

Step1: Calculate the SAD of each vertex. Label the vertex with the highest SAD as X_w and the vertex with the lowest SAD as X_b, the other as X_n .The SAD values are $SAD(X_w)$, $SAD(X_b)$, $SAD(X_n)$, respectively.

Step2: Calculate the centroid vertex X_c of all vertices excluding X_w: $X_c = (X_n + X_b)/2$

Step3: Reflect X_w through the centroid vertex X_c to get a new vertex X_r: $X_r = X_c + \alpha(X_c - X_n)$

Step4: If $SAD(X_r) < SAD(X_b)$, the reflection was in the correct direction, then expand the vector $(X_r - X_c)$ to the vertex $X_e : X_e = X_c + \gamma(X_r - X_c)$ and calculate $SAD(X_e)$ as well.

If $SAD(X_e) < SAD(X_b)$, replace X_w by X_e, and compose new three points $(X_b X_n X_e)$. Otherwise, replace X_w by X_r, and compose new three points $(X_b X_n X_r)$.

Step5: If $SAD(X_r) > SAD(X_n)$, contract the vector $(X_w - X_c)$ to the vertex $X_s : X_s = X_c + \beta(X_w - X_c)$, and calculate $SAD(X_s)$, replace X_w by X_s, and compose new three points $(X_b X_n X_s)$.

Step6: If $SAD(X_r) > SAD(X_w)$, reflection is failed, then replace X_w by the midpoint of X_w and X_b, replace X_n by the mid-point of X_n and X_b, and compose new three points.

3.5 MSPSO algorithm structure

The MSPSO algorithm for motion estimation is summarized as follows:

Step1: Select initial searching points of motion vectors and define initial parameters such as iteration number, constriction factor, coefficients of simplex method.

Step2: Calculate the velocity and position for each point.

Step3: Calculate the fitness function (SAD) for each point, and if early termination is not met, go to Step 4, otherwise, terminate.

Step4: Calculate $pBest_{id}$ and $gBest_d$ for all points.

Step5: Mutation operation is implemented. Then re-calculate $pBest_{id}$ and $gBest_d$ for all points

Step6: SM searching is implemented to find better points to replace worst points.

Step7: Update $pBest_{id}$ and $gBest_d$ for all points.

Step8: If the termination criterion is not met, go to step 3, otherwise, terminate.

Step9: Return the best point of the minimum SAD, which position is the position of motion vector.

4 Simulation results

In the simulations, select first 100 frames of six typical standard test sequences: Akiyo and News (smooth motion), Coastguard and Bus (middle motion), Stefan and Football (violent motion). The image format is CIF (352*288) and frame rate is 30f/s. 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).

4.1 Parameter selection and analysis

To evaluate the performance of each algorithm, we use the PSNR (peak signal to noise ratio) as a measure of the algorithm's search precision.

$$PSNR = 10\log_{10}\left[\frac{255^2}{MSE}\right] \quad (9)$$

MSE is the mean square deviation between reconstructive frame and original frame. Because search time is influenced by running platform and several other factors, we use the average searching points to evaluate the computational complexity of all algorithms.

The performance of MSPSO algorithm is greatly impacted by the different values of iteration number, population size and mutation number. In order to obtain suitable values of block matching algorithm, we carried out experiments and analysis for the selection of parameters.

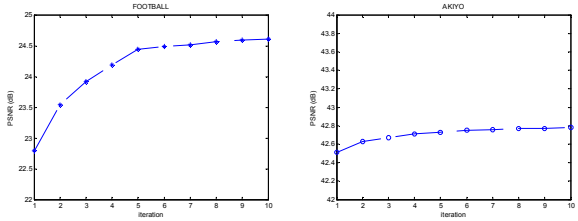
The purpose of fast block matching is guaranteed search accuracy while reducing computational complexity. The larger number of iterations, population size and mutation number will inevitably lead to higher search precision, but they will increase the search points. Therefore, we need to carefully select the appropriate parameter.

We select No. 20 frame of AKIYO and FOOTBALL sequences to experiment and analysis.

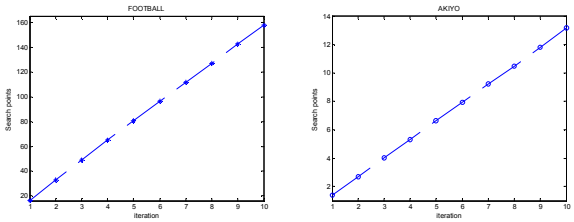
(1) Discuss of iteration number

Because iteration number has a great influence on the search points, so we discuss firstly the size of iteration number. Pre-defined initial population size is 10, PSO mutation number is 2.

As shown in Fig.3 that PSNR is gradually raised with the increase of iteration number. After reaching a peak point, PSNR is changed slowly. However, the search points are raised linearly all the way. Especially for FOOTBALL sequence, the search points when iteration number is 10 are twice than that iteration number is 5. Therefore, we select the appropriate iteration number according to the peak of PSNR, and the search points are not high. As shown in Fig3, we select the iteration number is 5.



(a)



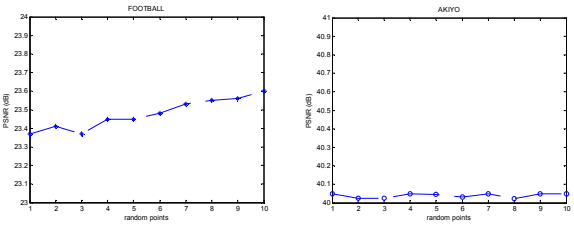
(b)

Fig.3 Iteration number effects on (a) PSNR (b) search points of Football & Akiyo sequences

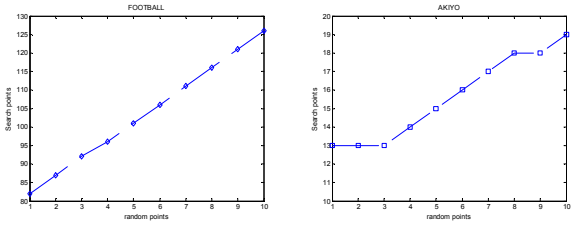
(2) Discuss of population size

When iteration number is 5, discuss the effect of population size on the performance of MSPSO algorithm. According to the temporal-spatial correlation of motion vectors, the initial fixed points are 5. Therefore, here we discuss the population size of the random point.

From Fig.4 we can see, with the increase of random points, PSNR value is almost unchanged, but the search points are raise rapidly, especially for FOOTBALL sequence. Therefore, in order to reduce the computational complexity, we select the random point is 1. So the population size is 6.



(a)



(b)

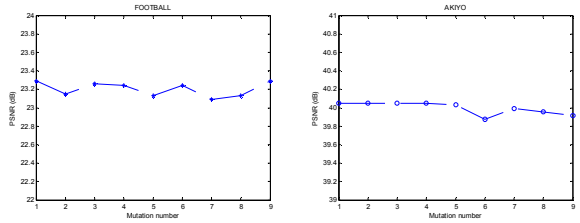
Fig.4 Population size effects on (a) PSNR (b) search points of Football & Akiyo sequences

(3) Discuss of mutation number

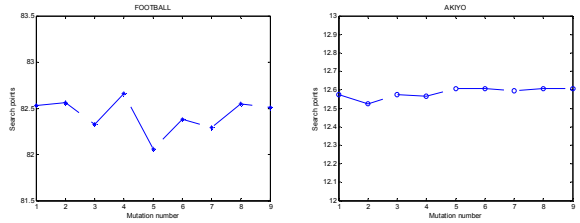
When iteration number is 5 and population size is 6, discuss the effect of mutation number on the performance of MSPSO algorithm.

From Fig.5 we can see, with the increase of mutation number, both PSNR values and search points only are fluctuated slightly. Therefore, in

order to reduce the computational complexity, we select the mutation number is 1.



(a)



(b)

Fig.5 Mutation number effects on (a) PSNR (b) search points of Football & Akiyo sequences

4.2 Simulation results and analysis

FS, FSS, DS, ARPS, are selected to compare with MSPSO algorithm. The other parameters of simulation of MSPSO are as follows: $c_1 = c_2 = 2$, $\kappa = 0.729$, $\alpha = 1$, $\beta = 0.5$, and $\gamma = 2$.

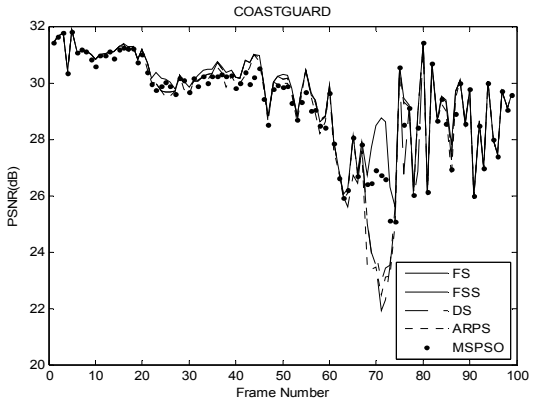
4.2.1 PSNR

The average PSNR values of various algorithms (FS, FSS, DS, APRS, MSPSO) and the difference values of PSNR with FS algorithm are listed in Tab.1.

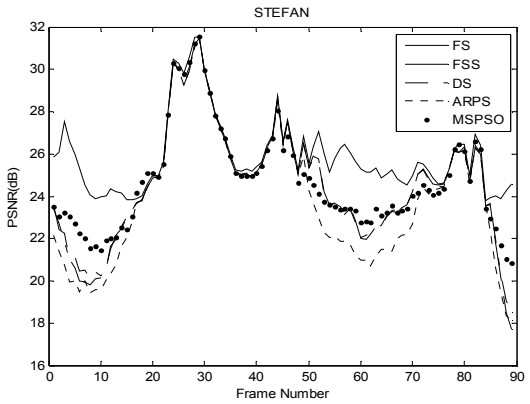
The results of Table 1 denote that the difference between PSNR of FSS, DS, ARPS and PSNR of FS is very small for Akiyo and News sequence which have smooth motion. When the motion is smooth most motion vectors are in the vicinity of the origin and unimodal error surface assumption is tenable. However, for the Stefan and Football sequence which have violent motion, the difference between PSNR of FSS, DS, ARPS and PSNR of FS is increased significantly. Because when motion is violent, motion vectors are far from the origin, the hypo-best peaks increase and the assumption based on unimodal error surface is invalid, the conventional fast algorithms are easily trapped in local minimum and global optimal solution cannot be obtained.

MSPSO which is a combination of global optimization ability of PSO with mutation and local optimization ability of SM, makes the particle has diversity and at the same time ensures the evolution convergence of the particles, which can avoid falling into local optimum, obtain the global optimum. As shown in Table 1, MSPSO algorithm performs better than FSS, DS, ARPS for the violent sequences and the fluctuation of PSNR difference between it and FS is slight.

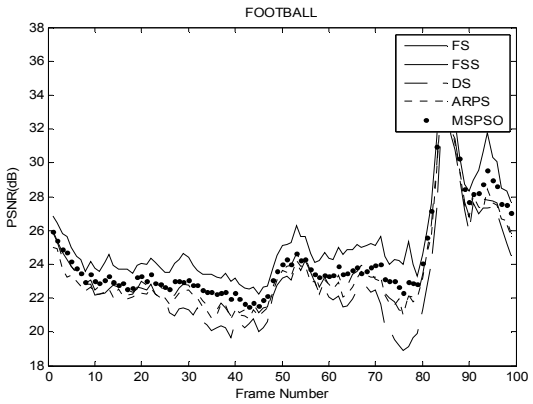
Fig.6 shows the PSNR differences between FS and each other algorithm of every frame of Coastguard, Stefan, Football sequences. It can be seen that the PSNR differences between FS and FSS, DS, ARPS have significant fluctuation, which makes the quality of video to fluctuate. And MSPSO not only have a better search accuracy than these fast algorithms but also reduce the significant fluctuation of video quality, especially for some part of the video sequence with violent motion, such as 50-70 frames of Stefan sequence and 77-82 frames of Football sequence. FSS, DS, ARPS algorithm is fallen into local optimum and PSNR of them decrease significantly. However, because of the addition of mutation and simplex method, MSPSO avoids local optimum effectively and the PSNR decreases slightly.



(a)



(b)



(c)

Fig.6 Comparison of PSNR of whole sequence: (a) Coastguard (b) Stefan (c) Football

4.2.2 Average searching points

In order to compare the computational complexity of various algorithms, each algorithm is executed 10 times for 100 frames of every sequence to get the average search points for motion estimation. FSS, DS and ARPS also adopts early termination strategy when calculate the number of search points, so the number obtained is slightly less than that without early termination strategy.The average searching points is shown in Table 2.

The average searching points of SPSO algorithms are much less than that of FS, and a little more than that of TSS and DS. The reason is that SPSO algorithm is the probabilistic searching algorithm and need to execute various evolution operations in each generation, while, TSS and DS only need to search few fixed number of points for each macro block.

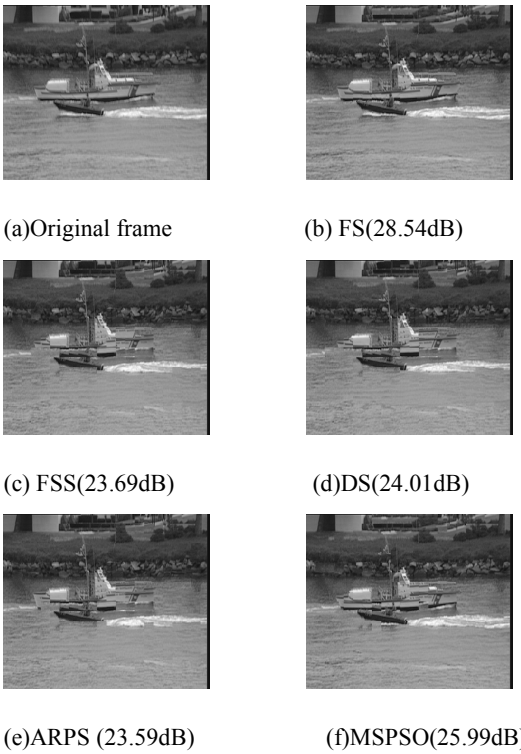
It can be seen from Table 2 that, the average searching points of FSS, DS and ARPS are fluctuated slightly for various sequence, however, the average searching points of MSPSO have a little high fluctuation for these sequences. ARPS has the least searching point but it has the least PSNR for the sequences with violent motion. For Akiyo with smooth motion, MSPSO has less searching point, which is only more than that of ARPS. As the motion intensity increase, the searching point of MSPSO is increased gradually. For Coastguard with medium motion, the searching point of MSPSO is about 3 times than that of FSS and DS. While for the sequences with violent motion like Football, the searching point of MSPSO is 4 times than that of FSS and DS. Despite this, searching point of MSPSO is much less than that of FS, less than one tenth that of FS.

Weighing the search performance and computational complexity, the search performance of MSPSO for various sequences improves significantly at the cost of the increasing of computational complexity. But for relative static sequences, the computational complexity of MSPSO is still less than that of FSS, DS, ARPS when ensure the search performance.

4.2.3 Comparison of the subjective image

In order to give the visual result, subjective image of the motion compensation are shown as follow.

Between FS, FSS, DS, ARPS and MSPSO algorithms, the motion compensation results of Coastguard No.70 frame and Stefan No.60 frame are shown in Fig. 7 and Fig. 8 we can see, FSS, DS, ARPS algorithms have obvious blocking effect in motion details, however MSPSO algorithm has significantly improvement, very close to the FS algorithm.



(a)Original frame (b) FS(28.54dB)

(c) FSS(23.69dB) (d)DS(24.01dB)

(e)ARPS (23.59dB) (f)MSPSO(25.99dB)

Fig.7 Coastguard No.70 frame: Comparison of subjective image of the motion compensation between FS, FSS, DS, ARPS, MSPSO

(a)Original frame (b) FS(25.40dB)

(c) FSS(21.99dB) (d)DS(21.65dB)

(e)ARPS (20.96dB) (f)MSPSO(22.29dB)

Fig.8 Stefan No.60 frame: Comparison of subjective image of the motion compensation between FS, FSS, DS, ARPS, MSPSO

5 Conclusions

In this paper, a new MSPSO algorithm for fast motion estimation is proposed. MSPSO algorithm incorporates SM technique and PSO algorithm, overcomes the local minima sticking problem. The experimental results show that MSPSO algorithm can provide accurate search matching and reduce the computational complexity of motion estimation. Consequently, MSPSO algorithm can obtain the fast and efficient search results with little sacrifice in terms of PSNR value and it is suitable to be applied in video coding systems.

Table 1 Comparisons of the average PSNR of FS， FSS，DS，ARPS and MSPSO (unit: dB)

Sequences	Akiyo		News		Flower		Coastguard		Stefen		Football	
Algorithms	PSNR	△	PSNR	△	PSNR	△	PSNR	△	PSNR	△	PSNR	△
FS	42.81	0	37.36	0	26.03	0	29.61	0	25.90	0	25.39	0
FSS	42.73	-0.08	37.16	-0.2	25.76	-0.27	29.21	-0.63	24.49	-1.50	22.96	-2.43
DS	42.79	-0.02	37.23	-0.13	25.98	-0.05	29.32	-0.29	24.53	-1.37	23.84	-1.55
ARPS	42.77	-0.03	37.17	-0.19	25.98	-0.05	29.19	-0.42	23.90	-2.00	23.54	-1.85
MSPSO	**42.79**	**-0.02**	**37.27**	**-0.09**	**25.99**	**-0.04**	**29.32**	**-0.29**	**24.70**	**-1.20**	**24.38**	**-1.01**

Table 2 Comparison of average searching points

Sequences	Akiyo		News		Flower		Coastguard		Stefen		Football	
Algorithms	Point	ratio	PSNR	△	PSNR	△	Point	ratio	Point	ratio	PSNR	△
FS	1024	100%	1024	100%	1024	100%	1024	100%	1024	100%	1024	100%
FSS	15.9	1.55%	16.1	1.57%	18.4	1.80%	18.6	1.81%	18.5	1.81%	22.1	2.16%
DS	12.3	1.20%	12.7	1.24%	15.8	1.54%	12.3	1.20%	17.2	1.68%	28.4	2.77%
ARPS	5.06	0.49%	5.51	0.54%	8.6	0.84%	9.99	0.98%	10.9	1.06%	21.7	2.12%
MSPSO	**8.62**	**0.84%**	**13.9**	**1.36%**	**40.8**	**3.98%**	**59.1**	**5.78%**	**67.7**	**6.61%**	**80.5**	**7.86%**

References:

[1] STILLER C, Estimating motion in image sequences: a tutorial on modeling and computation of 2d motion, IEEE Signal Process. Mag., Vol.16, No.4, 1999, pp. 70–91

[2] Y.-W. Huang, B.-Y. Hsieh, S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC, IEEE Trans. Circuits Syst. Video Technol., Vol.16, No.4, Apr. 2006, pp. 507–522.

[3] Lai-Man Po, Wing – Chung Ma, A novel four-Step search algorithm for fast block motion estimation, IEEE Trans. Circuits and Systems for Video Technology,Vol.6 , No.3, June 1996 ,pp. 313-317.

[4] S. Zhu, K.-K. Ma, A new diamond search algorithm for fast block-matching motion estimation, IEEE Trans. on Image Processing, Vol. 9, Feb. 2000, pp. 287-290.
[5] So H, Kim J, Cho W-K, Kim Y-S, Fast motion estimation using modified diamond search pattern, Electronics Letters, Vol.41, No.2, 2005, pp.62-63.
[6] Yao Nie, Kai-Khuang Ma., Adaptive rood pattern search for fast block-matching motion estimation, IEEE Trans. Image Processing, Vol.11, No.12,2002,pp.1442-1448.
[7] Jang-Jer Tsai, Hsueh-Ming Hang, Modeling of pattern-based block motion estimation and its Application, IEEE Transactions on Circuits and Systems for Video Technology, Vol.19, No.1, 2009,pp.108-113.
[8] Ka-Ho Ng, Lai-Man Po, Ka-Man Wong, et al. , A search patterns switching algorithm for block motion estimation, IEEE Transactions on Circuits and Systems for Video Technology, Vol.19,No.5,2009,pp.753-759.
[9] Guang-yu DU, Tian-shu HUANG, Li-xin SONG, et al., A novel fast motion estimation method based on particle swarm optimization, The Fourth International Conference on Machine Learning and Cybernetics,2005, pp.5038-5042.
[10] K. M. Bakwad, S.S. Pattnaik , B. S. Sohi , et al. , Small population based modified parallel particle swarm optimization for motion estimation, 16th International Conference on Adavanced Computing & Communication (ADCOM 2008), 2008, pp.367-373.
[11] Yuan Xuedong, Shen Xiaojing, Block matching algorithm based on particle swarm optimization for motion estimation, The 2008 International Conference on Embedded Software and Systems (ICESS2008),2008, pp.191–194.
[12] D. Ranganadham, Pavankumar gorpuni, An efficient bidirectional frame prediction using particle swarm optimization technique, International Conference on Advances in Recent Technologies in Communication and Computing, 2009, pp.42-46
[13] J. Kennedy , R. C. Eberhart, Particle swarm optimization, in Proc. IEEE Int. Conf. Neural Netw., Perth, Australia, vol. 4, 1995, pp. 1942–1948.
[14] J. Kennedy, R. C. Eberhart, Y. H. Shi, Swarm Intelligence. San Mateo, CA: Morgan Kaufmann, 2001.
[15] R. A. Krohling ,L. dos Santos Coelho, Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems, IEEE Trans. Syst., Man, Cybern. B, Cybern, Vol. 36, No. 6, Dec. 2006, pp. 1407–1416.
[16] S.-Y. Ho, H.-S. Lin, W.-H. Liauh, S.-J. Ho, OPSO: Orthogonal particle swarm optimization and its application to task assignment problems, IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, Vol. 38, No. 2, Mar. 2008, pp. 288–298.
[17] Zhi-Hui Zhan, Jun Zhang, Yun Li, et al., Adaptive particle swarm optimization,IEEE Transactions on Systems, Man, And Cybernetics-Part B:Cybernetics, Vol.38, No.6,2009, pp.1362-1381.
[18] B. Bunday, Basic Optimization Methods, Edward Arnold, London, UK, 1984.
[19] NEDLER, J.A., MEAD, R.,A simplex method for function minimization, Comput. J., 1965, 7, pp. 308-313
[20] M. Rehan, P. Agathoklis, A. Antoniou, Flexible triangle search algorithm for block based motion estimation, in Proceedings of the IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing (PACRIM '03), Vol. 1, Victoria, BC, Canada, August 2005,. pp. 233–236
[21] Mohamed Rehan, Pan Agathoklis, Andreas Antoniou, Flexible triangle search algorithm for block-based motion estimation. EURASIP Journal on Advances in Signal Processing,2007, pp.1-14