2009 International Conference on Advances in Recent Technologies in Communication and Computing
2009, Article number 5328092, Pages 42-46, DOI: http://dx.doi.org/10.1109/ARTCom.2009.87

An efficient bidirectional frame prediction using particle swarm optimization technique

D. Ranganadham
All India Radio & Doordarshan
Bhubaneswar,India
rangi.d@gmail.com

Pavankumar gorpuni,
G. Panda, Senior Member IEEE,
Department of ECE
NIT ROURKELA
Rourkela, India
pavankumar.mtech@gmail.com,
ganapati.panda@gmail.com

***Abstract*—This paper presents a Novel Bidirectional motion estimation technique, which is based on the Particle swarm optimization algorithm. Particle swarm optimization (PSO) is a population based optimization technique which has the potentiality to avoid local minima solution which is usually encountered by the traditional block matching algorithms (BMA) such as the three step search (TSS) and the diamond search (DS). To speed up the search, static macro blocks are found in our method, which is particularly beneficial to those video sequences containing small motion contents. Skipping such static macro blocks from processing can save the computation time and memory also. In the proposed method each time we are finding the best matching macro block in two frames at a time so we can reduce the number of error function calculations and it is faster than if we apply PSO technique to find forward motion vector and backward motion vector separately. The proposed method is applied to a number of benchmark video sequences and the results are compared with those obtained by applying the existing methods. Simulation results shows that the proposed algorithms gives the close match of PSNR values when compared to joint search algorithm with DS. Thus, PSO algorithm for Bidirectional motion estimation is empirically given to reduce computational complexity.**

***Keywords*—B frames, Average mean square prediction, Macroblock, Particle swarm optimization, Motion vector.**

I. INTRODUCTION

Image sequences usually contain a high degree of temporal redundancy that can be exploited for coding and processing purposes. Motion estimation and compensation is a powerful means of exploiting this redundancy and is used in most advanced video coders, including the MPEG and H.263 video coding standards [1]. MPEG provides for up to three types of frames called I, P and B frames. The intra frame, or I frame, serves as a reference for predicting subsequent frames. I frames, which occur on an average of one out of every nine to fifteen frames, only contains information presented within itself. P Frames are predicted from information presented in the nearest preceding I or P frames. The bi-directional B frames are coded using prediction data from the nearest preceding I or P frame and the nearest following I or P frame.

B pictures are pictures in a motion video sequence that are encoded using both past and future pictures as references. The prediction is obtained by a linear combination of forward and backward prediction signals usually obtained with motion compensation. However, such a superposition is not necessarily limited to forward and backward prediction signals [2]. Bi-directional frame prediction (B frame coding) uses a past frame and a future frame as two reference frames for prediction. B frame coding provides a number of significant advantages that of occlusion and scene changes. Occlusion refers to the covering/uncovering of a surface due to three-dimensional (3-D) rotation and translation of an object that occupies only part of the field of view. It occurs quite often in real-world images. For example, objects move in front of other objects, objects move toward the camera, cameras zoom, and objects rotate. If only two frames are used at such regions, the motion estimation (ME) algorithm will not be able to find a good estimate of the underlying motion at these regions because there is no corresponding feature in the other frame to match. In addition, this bad motion estimate could affect other surrounding motion vectors due to the spatial smoothness constraints applied in various ME algorithms [3]. Despite the advantages, B frame coding introduces an extra delay in the encoding process, which has become a problem in applications such as teleconferencing.

In contrast to unidirectional motion estimation (ME), which finds a single motion vector for each macro block, the bidirectional motion estimation computes a pair of motion vectors (the forward vector, VF, and the backward vector, VB) by searching for the matching macro blocks in the past and the future reference frames [4] as shown in Fig. 1. Therefore, Bidirectional ME forms a major computation bottleneck in video processing applications such as the detection of noise in image sequences. The performance of bidirectional motion estimation could be improved by

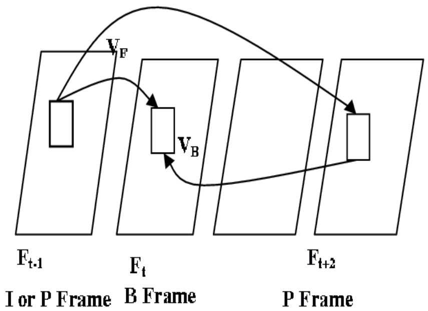

Fig 1. Bidirectional frame prediction

exploiting temporal correlation between the frames. One solution is to recursively perform a joint estimation of forward and backward vectors [5]. Similarly, [6] computes one of the vectors first, while the other one is derived by reversing the previously obtained vector and using it as an initial seed to search within a small region. In [6], the vector propagation algorithm postpones the forward motion vector field computation so that it can be used as the backward motion vector field. An approach to combine the motion vector tracking [7] with spatial motion prediction [8] is given in [9]. Although these methods differ in implementation, all of them compute at least one motion vector for each B-frame macro block and then use it for prediction of the other motion vector.

The rest of the paper is organized as follows. Section II introduces the Particle Swarm Optimization (PSO) and we propose the PSO based bidirectional frame prediction scheme in Section III. Simulation results and analysis on four video sequences are given in Section IV. Section V concludes with the findings of the paper.

II. Particle Swarm Optimization

A. Particle Swarm Optimization

Particle swarm algorithm [10] is a kind of evolutionary algorithm based on swarm intelligence. Each potential solution is considered as one particle, and these particles are distributed stochastically in the high-dimensional solution space in the initialization period of the algorithm. Through following the optimum discovered by itself and the entire group, each particle periodically updates its own velocity and position.

$$v_{id}(t+1) = w * v_{id}(t) + c_1 * rand * (p_{id} - x_{id}(t)) + c_2 * rand * (p_{gd} - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) \quad (2)$$

Where, N is the number of particles and D is the dimensionality; $V_{id} = (v_{i1}, v_{i2}, \ldots v_{iD})$ is the velocity vector of particle i which decides the particle's displacement in each iteration. Similarly, $X_{id} = (x_{i1}, x_{i2}, \ldots x_{iD})$ is the position vector of particle i which is a potential solution in the solution space. $[-v_{max}, v_{max}]$ is the range of the velocity vector and $[-x_{max}, x_{max}]$ is the range of position vector .The quality of the solution is measured by a fitness function; w is the inertia weight which decreases linearly during a run; c1, c2 are both positive constants, called the acceleration factors which are generally set to 2.0; rand1 (.) and rand2 (.) 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 termination criterion for iterations is determined according to whether the presetting maximum generation or a designated value of the fitness is reached.

B. PSO based unidirectional motion estimation

The traditional fast BMA assumes that the error function has only one global optimum solution and the error monotonically decreases as the search point moves towards it. Since the two assumptions are not valid in the real world problem the performance of BMA is not satisfactory. Particle swarm optimization (PSO) was originally proposed by Kennedy and Eberhart in 1995. Unlike genetic algorithm, PSO does not need genetic operators such as crossover and mutation. Thus it has advantages of easy implementation, fewer parameters adjustment, strong capability to escape from local optima and rapid convergence characteristic.

Evolutionary computing techniques such as genetic algorithm (GA) [11], particle swarm optimization (PSO) [12] have been successfully applied to solve many motion estimation problems. These methods are suitable for achieving global optimal solution, which traditional fast BMAs are not able to obtain easily. The GA requires some key parameters such as population size, probability of mutation, probability of crossover, etc. for yielding acceptable performance. In contrast the PSO involves simple computation and has been successfully applied to unidirectional motion estimation.

Even when we apply the particle swarm optimization technique to solve motion estimation we need to apply these technique separately twice for Forward motion vector V_F and backward motion vector V_B and then we need to follow some measures to decide which motion vector is giving least error and which frame is giving that result, but this is a time consuming process because motion vector overhead will be huge. Our idea is not to find the forward and backward motion vectors individually but to find the minimum matching Macro block at each time when PSO is finding for a minimum matching block, so it will reduce the number of computations involved in finding out the minimum matching point.

III. Praposed Algorithm for Bidirectional Frame Prediction

A. Static Macroblock Prediction

As more than 70% of the MacroBlocks (MB) of real world video sequences are static and hence do not need the remaining search [13]. Therefore significant reduction of computation is possible if we predict the static macro blocks before starting motion estimation procedure and the remaining search would be faster and save memory. We first calculate the matching errors sum of absolute difference (SAD) between the MB in the current frame and the MB at the same location in the reference frame and then compare it to a predetermined threshold, Δ. If the matching error is smaller than Δ we consider the MB static which do not need any further motion estimation, and return a [0, 0] as its motion vector (MV). Threshold value for each test video sequence correspondingly based on data obtained in experiments as shown in Table 1

B. *Intial Particles Positions and Size*

Block-based matching algorithms consider each frame in the video sequence formed by many non overlapping small regions, called Macro block (MB) which is often square-shaped and with fixed-size (16×16 or 8×8). We put four particles in a cross shape with size one (size refers to the distance between any vertex point and the center-point) in the adjacent MBs and four particles in a cross shape with size two, and then rotate it by angle Π/2 as shown in Fig.2. With two cross shapes in different sizes, we try to balance the global exploration and local refined search in order for broader searching space as well as higher matching accuracy. Moreover, we distribute particles equally in all directions (8 particles in 8 directions) with a view to find the matching MB in each direction with equal possibility.

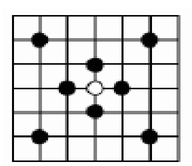

Fig.2 Particles initial position

C. *Algorithm Steps*

The proposed algorithm can be summarized in the following algorithm steps

- Perform block matching algorithm based on PSO.
- Static Macro block prediction
- Initial particles positions
- Each time find the minimum matching error (SAD) point in the past frame (SAD_P) and the future frame (SAD_F) with the current frame as shown in Fig.3
- Take the minimum out of both matching error (SAD), this one we are taking as the Cost function of our algorithm.
- For each generation we are getting the minimum matching error point in the two reference frame at a time.
- Until it reaches the stopping criteria it will continue the above steps. Fixed stopping criteria is adopted
- Save the final motion vector point for motion compensation.

Since we are performing the Block matching procedure at a time in two reference frames, our objective function is to minimize the minimum of the two matching errors between two frames.

$$\text{Cost function} = \min(SAD_P, SAD_F)$$

Where SAD_P and SAD_F are the sum of absolute difference of the past frame and future frames. Here for stopping criteria we adopt the fixed-iteration method in this paper for reducing the computational cost.

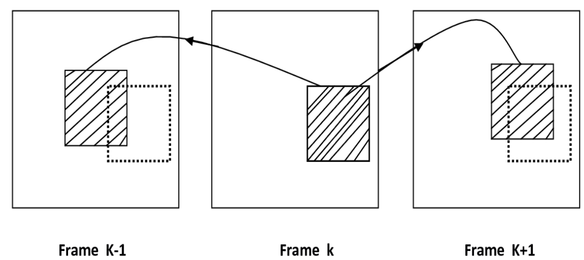

Fig.3. Bidirectional search for best match in two frames

IV. EXPERIMENTS AND RESULTS

A. *Motion Estimation Parameters*

We divide a whole image frame into 16 × 16 MBs in the simulation that is N=16 and the size of search window was set as 15×15. Threshold for each test video sequence correspondingly based on data obtained in experiments shown in Table 1. This threshold value is not fixed, may vary depending on your video sequences. We do not restrict the range of candidate matching MBs rigidly by a search window P. Instead, through the fixed-iteration and the setting of max velocity, particles search for the matching MB in an area more flexible and adaptable. The inertia weight w decreases linearly from 0.9 to 0.4 during a PSO run and two Acceleration coefficients C1 and C2 can also control how far a particle will move in a single iteration. Typically, both of them are initialized as 2.0 [14].Maximum generations are 5 and the maximum velocity is 10.

B. Results and analysis

The performance of the proposed Bidirectional motion estimation block matching algorithm based on Particle Swarm Optimization is evaluated in terms of Average mean square prediction error (AMSPE), Average Search points/Macro block and peak signal-to-noise ratio (PSNR) per frame of the reconstructed video sequence is computed for quality measurement and static Macro block prediction procedure to be done before starting of block matching procedure.

Experiments conducted over four video sequences demonstrate that the proposed technique is superior to the existing bi-directional motion compensation methods as are shown in tables for different video sequences schemes. The prediction error is averaged over 100 frames. The motion estimation for our bi-directional prediction coding is conducted between a B-frame and a past I or P and a future frame P. since the joint search at all the locations within the search windows in the previous and the future frame is computationally very expensive. So the proposed technique will do the motion vector search in both frames at a time. When finding out the matching error between frames each time one is considered as the local best position for each particle position. After the limited number of generations we will get the global minimum point in both of the frames. So we reduced a lot of number of error function calculations .Due to the minimum computational Cost, we choosen Summed Absolute Difference (SAD) as the error function.

As widely adopted, we measure the amount of computation in terms of Average Search points/macro block and the quality of compensated video sequence by Computation criterion and Peak Signal-to-Noise Ratio (PSNR) and Average mean square prediction error (AMSPE) for each frame. The pattern of the group of picture (GOP) IBBPBBPBBI . The evaluation is based on the AMSPE. The results reported in the table are based on the average mean square prediction error (AMSPE). The AMSPE values for all the four video sequences namely News, Mother & Daughter, Akyio, Silent are noted in a tabular form 2 against the Joint search algorithm based on Diamond search and Particle Swarm Optimization. To find the fastness of the algorithm we find out Average Search points/macro block for all the four video sequences and are reported in the tabular form 3 against the Joint search algorithm based on Diamond search and Particle Swarm Optimization. As it can be seen that from the tables the proposed bidirectional algorithm is giving less prediction error and the number of search point per each frame are less.

For comparison we find out the quality between the reconstructed video sequences is computed by PSNR of all bidirectional frames and shown in a figure 4 for NEWS video sequence against the Joint search algorithm based on Diamond search and Particle Swarm Optimization. As it can be seen from these two tables and the PSNR (dB) values, the proposed method employing the bi-directional motion vectors requires less number of bits for a fixed AMSPE or produces a better prediction error for a given bit budget. These experimental results are obtained for the video sequences, It is quite clear that the proposed method can significantly reduces the computational complexity involved in the Bidirectional frame prediction and also least prediction error in all video sequences.

V. CONCLUSION

Bidirectional ME forms a major computation bottleneck in video processing applications such as the detection of noise in image sequences, interpolation/ prediction of missing data in image sequences and de-interlacing of image sequences. The proposed novel bidirectional motion estimation algorithm which can effectively reduces the number of operations in Block matching motion estimation without sacrificing the quality of the results. Proposed bidirectional algorithm is giving less prediction error and the number of search point per each frame are less. In addition, skipping those static macro blocks from processing can reduce the computational cost of the algorithm. Simulation results shows that the proposed algorithms gives the close match of PSNR values when compared to joint search algorithm with DS and an acceptable degree of drop when compared to joint search DS in some highly dense motion sequences. Moreover PSO just consumes a few lines of codes due to its simplicity which makes the PSO algorithm attractive for hardware implementation. In the future, variants of PSO might be applied to strengthen the global searching ability and to speed up the search and to avoid being trapped in local minima.

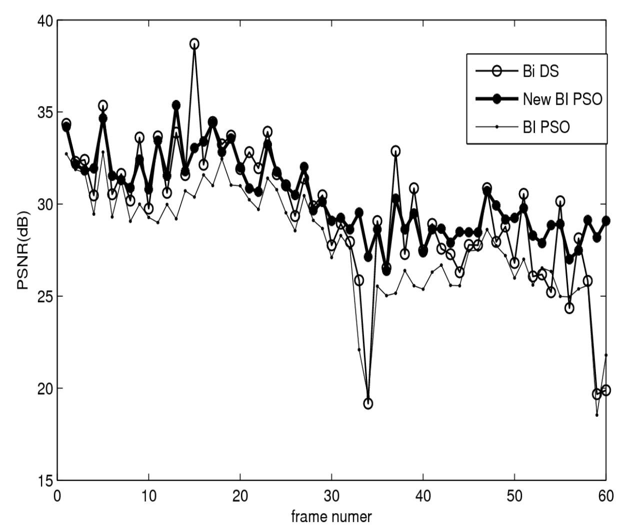

Fig. 4 frames PSNR (dB) comparison of NEWS video

TABLE I. THRESHOLD FOR FIVE TEST VIDEO SEQUENCES

Sequence	Assumed Threshold	
	Format	***Threshold***
Akiyo	QCIF	384
Mother &Dau.	QCIF	250
News	QCIF	250
Silent	QCIF	300

TABLE II. AVERAGE MEAN SQUARE PREDICTION ERROR

Sequence	AMSPE		
	BI-DS	***BI-PSO***	***NEW BI PSO***
News	125.1636	148.0847	80.3798
Mother	32.4141	47.3537	29.5195
Akiyo	11.8697	20.0268	16.3192
Silent	76.9223	98.5123	51.4913

TABLE III. AVERAGE SEARCH POINTS/MACROBLOCK

Sequence	Average Search points/Macro block		
	BI-DS	***BI-PSO***	***NEW BI PSO***
Akiyo	21.6463	14.1150	4.6245
Mother & daughter	21.3988	15.2325	9.5189
News	19.7825	11.9449	9.3096
Silent	19.9050	9.7298	10.2022

REFERENCES

[1] A. M. Tekalp, *Digital Video Processing*. Upper Saddle River, NJ: Prentice-Hall, 1995.

[2] B. Girod, “Efficiency analysis of multihypothesis motion-compensated prediction for video coding,” *IEEE Trans. Image Processing*, vol. 9, pp. 173–483, Feb. 2000.

[3] S. L. Iu, “Robust estimation of motion vector fields with discontinuity and occlusion using local outlier rejection,” *SPIE 2094,* pp. 588–599, Dec. 1996.

[4] M. K. Kim and J. K. Kim, “Efficient motion estimation algorithm for bi-directional prediction scheme,” *Electron. Lett.,* vol. 30, pp. 632–633, Apr. 14, 1994.

[5] S.W. Wu and A.Gersho, “Joint estimation of forward/backward motion vectors for MPEG interpolative prediction”, *IEEE Trans.Image Proc*., vol.3, pp.684-687, Sept. 1994.

[6] S.Katra, M-N. Chong, “Bidirectional motion estimation via vector propagation”, IEEE Trans. on CAS for Video Techn., Vol.8, no.8, pp.976-987, Dec.1998.

[7] J. Yeh, M. Vetterli, M. Khansari, “Motion compensation of motion vectors”, *Proc. IEEE ICIP*, vol.1, pp.574-578, 1995.

[8] A.M.Tourapis, E.Wu, S.Li, “Direct mode coding for bipredictive pictures in the JVT standard”, IEEE ISCAS,vol..2,pp.700-703, 2003.

[9] X.Li, Y.Lu, D.Zhao, W.Gao, S.Ma, “Enhanced direct coding for bipredictive pictures”, Proc. IEEE ISCAS, vol.3, pp.785-788, 2004

[10] J.Kennedy and R.C.Eberhart. Particle swarm optimization. In Proc. IEEE International Conference on Neural Networks, Perth, Australia, 1995.

[11] S. Li, W.-P Xu, N.-N Zheng, H. Wang, “A novel fast motion estimation method based on genetic algorithm,” *ACTA* ELECTRONICA SINICA, vol. 28, No. 6, pp. 114-117, June 2000.

[12] Xuedong Yuan, Xiaojing Shen, “Block Matching Algorithm Based on Particle Swarm Optimization for Motion Estimation,” *The 2008 International Conference on Embedded Software and Systems (ICESS2008)*, pp. 191-197.

[13] Y. Nie and K.-K. Ma, “Adaptive rood pattern search for fast block matching motion estimation,” IEEE Trans. Image Proc., vol. 11, no. 12, pp.1442-1449,Dec.2002.

[14] R.. C. Eberhart, P. Simpson, and R.. Dobbins, *Computational Intelligence PC Tools: Academic*, 1996, ch. 6, pp. 212–226.