International Journal of Electronics Signals and Systems

Volume 3 | Issue 3 Article 12

January 2014

AN EFFICIENT MOTION ESTIMATION ALGORITHM BASED ON PARTICLE SWARM OPTIMIZATION

ASHA ELIZABETH JACOB
Dept. of Electronics and Communication Engineering, Karunya university, Tamilnadu, India, ashaiby7@gmail.com

IMMANUEL ALEX PANDIAN
Dept. of Electronics and Communication Engineering, Karunya university, Tamilnadu, India, immans@karunya.edu

Recommended Citation

JACOB, ASHA ELIZABETH and PANDIAN, IMMANUEL ALEX (2014) "AN EFFICIENT MOTION ESTIMATION ALGORITHM BASED ON PARTICLE SWARM OPTIMIZATION," *International Journal of Electronics Signals and Systems*: Vol. 3 : Iss. 3 , Article 12.
DOI: 10.47893/IJESS.2014.1168
Available at: https://www.interscience.in/ijess/vol3/iss3/12

AN EFFICIENT MOTION ESTIMATION ALGORITHM BASED ON PARTICLE SWARM OPTIMIZATION

ASHA ELIZABETH JACOB[1] & IMMANUEL ALEX PANDIAN[2]

[1,2]Dept. of Electronics and Communication Engineering, Karunya university, Tamilnadu, India
E-mail: ashaiby7@gmail.com, immans@karunya.edu

Abstract- The PSO algorithm reduce the search points without the degradation of the image quality. It provides accurate motion estimation with very low complexity in the context of video estimation. This algorithm is capable of reducing the computational complexity of block matching process. This algorithm maintains high estimation accuracy compared to the full search method. The critical component in most block-based video compression system is Motion Estimation because redundancy between successive frames of video sequence allows for compression of video data. These algorithms are used to reduce the computational requirement by checking only some points inside the search window, while keeping a good error performance when compared with Full Search and Diamond search algorithm. This algorithm should maintain high estimation accuracy compared to the Full search method and Diamond search algorithm. Here by using the PSO algorithm could get a high accuracy in the block-based motion estimation.

Keywords*- Motion estimation; full search method; and particle swarm optimization.*

I. INTRODUCTION

In block based motion estimation, image frames in a video sequence are divided into blocks. There are two frames & it is divided into blocks. In each block one search for the best matching searches will be taken compared to the reference frame so that the matching metric will be reduced. In this way two frames compression can be done by taking the best match by the best matching searches. Among early block Motion Estimation algorithms, fixed search pattern algorithms are the most famous. These algorithms are used to reduce the computational requirement by checking only some points inside the search window, while keeping a good error performance when compared with Full Search algorithm. To reduce the search points without the degradation of the image quality [1]. This algorithm be capable of reducing the computational complexity of block matching Process.Compared to the full search method this algorithm maintains high estimation accuracy .Three step Search [2] algorithm is a fixed search algorithm and it is one of the most popular Block Matching Algorithms. For stationary or quasi stationary blocks in the three step search algorithm it will easily lead the search to be trapped into a local minimum so for that an efficient three step search is used.It is based on MV distribution of real world video sequences [3].It consists of two search Patterns in which the first pattern, called large diamond search pattern (LDSP) comprises nine checking points and form a diamond shape. The second pattern consists of five checking points that make a small diamond pattern (SDSP). Great interest has been devoted to the study of different approaches in video compressions.

The high correlation between successive frames of a video Sequence makes it possible to achieve high coding efficiency by removing the temporal - redundancy. Motion estimation (ME) and motion compensation techniques are an important part of most video encoding.

The most popular motion estimation and motion compensation method has been the block based motion estimation, which uses a block matching algorithm and the best matched block from a reference frame. The new algorithm uses a small cross-shaped search patterns in the first two steps to speed up the motion estimation blocks. Experimental results show that [4] new cross-diamond search algorithm could achieve higher computational reduction as compared with Diamond Search and Cross Diamond Search while similar prediction accuracy is maintained, and it is suitable for videoconferencing sequences. The video frames are partitioned into blocks of pixels in block-based motion estimation. Each block is estimated from a block of equal size in the reference frame. Specifically, for each block in the current frame, we search for a best matching block within a searching window in the reference frame. Block matching is an optimization problem and it has the goal of finding the best matching block within a specific search space. [5]PSO method is used to solve the block matching problem. To reduce the complexity and improve the performance of block matching algorithm, by selecting the initial individuals based on fixed points and random points.SM searching after finishing each iteration of PSO is used. SPSO can achieve better PSNR value than block matching algorithm [6]. In block matching based on PSO, a swarm of particles will fly through a region corresponding to a searching window in the reference frame, in any position as the particles correspond to candidate block in the reference frame, and it can be indexed by the horizontal and vertical coordinates of the center pixel of the candidate blocks. Therefore the dimension of the search space is two. In the following, we present a PSO-based block matching motion estimation algorithm, with a set of strategies introduced to speed up the motion search and provide high motion estimation accuracy**.**

II. MOTION ESTIMATION

The whole idea behind motion estimation based video compression is to save on bits by sending JPEG encoded difference images which inherently have less energy and can be highly compressed as compared to sending a full frame that is JPEG encoded. It should be noted that the first frame is always sent full, and so are some other frames that might occur at some regular interval.

The standards do not specify this and this might change with every video being sent based on the dynamics of the video [7]. The most expensive and resource hungry operation in the entire compression process is motion estimation. Hence, this field has seen the highest activity and research interest in the past two decades.

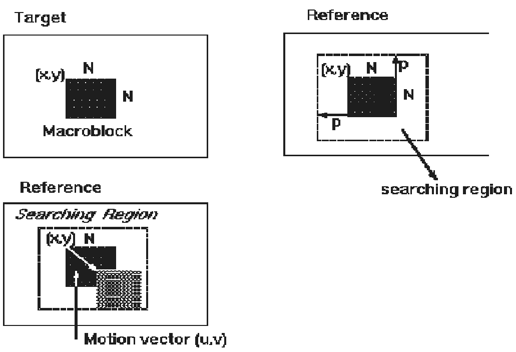

Fig1: Motion Estimation

In this section, we are concerned with the "projected motion" of 3-D objects onto the 2-D plane of an imaging sensor. By motion estimation, we mean the estimation of the displacement (or velocity) of image structures from one frame to another in a time sequence of 2-D images. To reduce the bit-rate of video sequences by removing temporal redundancy, motion estimation techniques have been developed.

For this reason, low computational complexity motion estimation algorithms are viable solutions. If a prior knowledge about the motion of the current block is available before the motion estimation then a better starting point for the search of a motion vector can be selected and also the computational complexity will be reduced. In this paper, we present a block matching algorithm based on temporal correlations of consecutive image frames that defines the search pattern and the location of initial starting point to reduce computational complexity.

Currently, great interest has been devoted to the study of different approaches in video compressions. The high correlation between successive frames of a video sequence makes it possible to achieve high coding efficiency by removing the temporal redundancy. The most popular motion estimation and motion compensation method has been the block based motion estimation, which uses a block matching algorithm to and the best matched block from a reference frame

FS can achieve optimal performance by examining all possible points in search area of the reference frame. However, FS is very computationally intensive and it can hardly be applied to any real time applications. Hence, it is inevitable to develop fast motion estimation algorithms for real time video coding applications. Regardless of the characteristic of the motion of a block, since this class of BMAs does not have any information on the motion of the current block, they use the origin of the search window as the starting point. To improve the accuracy of the existing BMA algorithms, the motion correlation between successive frames is used to predict an initial starting point that reacts the current block's motion trend. Because a properly predict initial starting point makes the global optimum closer to the predicted starting point, it increase the chance of ending the optimum or near-optimum motion vector with less search points.

III. BLOCK MATCHING ALGORITHM FOR MOTION ESTIMATION

In block-based motion estimation, the video frames are partitioned in non-overlapping blocks of pixels. Each block is predicted from block of equal size of that of the reference frame. Specifically, for each block in the current frame, we search for a best matching block within a searching window in the reference frame, which minimizes a certain matching metric. Block matching is essentially an optimization problem, with the goal of finding the best matching block within a search space. Matching of (all/some) pixels of current block with the candidate block in search area is performed.

Prediction of Motion Vectors is usually performed to gain an initial guess of next motion vector. This reduces the computational burden. In block matching algorithms, each macro block (8×8 pixels) in the new frame is compared with shifted regions of the same size from the previous frame, and the shift which results in the minimum error is selected as the best motion vector for that macro-block. Block motion estimation algorithms are widely adopted by video coding standards, mainly due to their simplicity and good distortion performance. Using block motion estimation, a video frame is divided into non-overlapping block of equal size and the best matched block is determined from reference frames to that block in the current frame within a predefined search window.

Full search which exhaustively evaluates all possible candidate blocks within the search window. It has been estimated that the computation of FS could consume up to 70% of the total computation of the video encoding process. To tackle this problem, many fast algorithms have been developed to speed up the motion estimation process. Among them, the most popular class is the block-matching algorithm using a fixed set of search point patterns, which is mainly due to very high speed up ratio in motion estimation process.

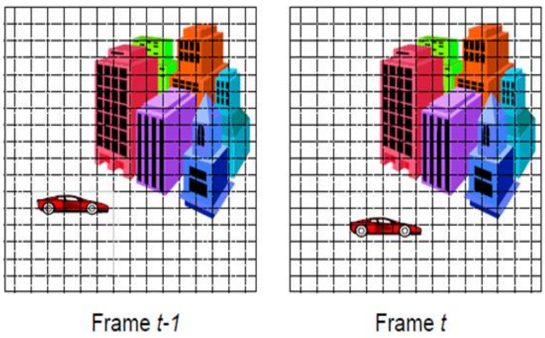

Fig 2: Block Matching Algorithm for motion estimation

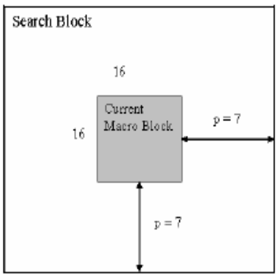

Fig 3:-Block matching a macro block of side 16 pixels and a search parameter p of size 7 pixels.

IV. ALGORITHMS USED

A. Full Search Algorithm

All candidates within search window are examined. It has good accuracy and it is used to find best match. Large amount of computation are required here: There are (2p+1)2 matches, and 16x16 MAE for each match. Impractical for real-time applications. In order to avoid this complexity; we should reduce search positions Fast Block Matching Algorithms.

B. Diamond Search Algorithm:-

It Start with 9 points creating a large diamond with center of search window in center. Find the best match, we must repeat with this best match as center. If the best match is in the center, then we use 5 points creating small diamond then we should find the best match the search starts with the LDSP and it is used repeatedly until the minimum BDM point lies on the search center. The search pattern is then turned to SDSP. The position yielding minimum error point is taken as the final MV.

C. Particle Swarm Optimization (PSO)

In block-based motion estimation, the video frames are partitioned in non-overlapping blocks of pixels. Thus one can readily apply the PSO method to solve the block matching problem. In block matching based on PSO, [8] a swarm of particles will fly through a region corresponding to a searching window in the reference frame, with any position traversed by the particles being corresponding to a candidate block in the reference frame, which can be indexed by the horizontal and vertical coordinates of the center pixel of the candidate blocks. Therefore the dimension of the search space is two. In the following, we present a PSO-based block matching motion estimation algorithm, with a set of strategies introduced to speed up the motion search and provide high motion estimation accuracy. A “swarm” is an apparently disorganized collection (population) of moving individuals that tend to cluster together while each individual seems to be moving in a random direction.

Particle swarm optimization applies to concept of social interaction to problem solving .It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution. Particle is treated as a point in a D-dimensional space which adjusts its “flying” according to its own flying experience as well as the flying experience of other particles. Each particle keeps the track of its coordinates in the problem space which is the best solution (fitness) it has achieved. This value is called pbest. Another best value that is tracked is the best value obtained by any particle in the neighbors of the particle so far. This value is called gbest.

The PSO concept consists of changing the velocity of each particle toward its pbest and the gbest position at each time. A “swarm” is an collection (population) of moving individuals that tend to cluster together while each individual seems to be moving in a random direction. Particle swarm optimization (PSO) applies to concept of social interaction to problem solving. It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution. In PSO there is no selection operation. All particles in PSO are kept as members of the population through the course of the run.

PSO is the only algorithm that does not implement the survival of the fittest. No crossover operation in PSO. The PSO concept consists of changing the velocity of each particle toward its pbest and the gbest position at each time .A “swarm” is a collection (population) of moving individuals that tend to cluster together while each individual seems to be moving in a random direction. Particle swarm optimization (PSO) applies to concept of social interaction to problem solving.

It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution. In PSO there is no selection operation. All particles in PSO are kept as a member of the population through the course of the run.PSO is the only algorithm that does not implement the survival of the fittest. No crossover operation in PSO. The idea of PSO is to change the velocity of each particle towards its pbest and gbest locations at each time step.

Each particle is treated as a point in the D-dimensional problem space.[9].The ith particle is represented as $X_1 = (x_{i1}, x_{i2}...., x_{iD})$ .The best previous position of the ith particle is recorded and represented as the position as $P_1 = (p_{i1}, p_{i2}...., p_{iD})$. The particles are initialized with random particles. The index of the best particle among all the particles in the population is represented by the symbol g.The rate of position change for particle i is represented as $V_1 = (v_{i1}, v_{i2}...., v_{iD})$.
The particles are manipulated according to[8] the following equations:

$$V_i(t+1) = wV_i(t) + c_1 r_1 [P_i(t) - X_i(t)] + c_2 r_2 [P_g(t) - X_i(t)] \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

Where i is the index of the particle, i = 1, 2. . . M; w the inertia weight; c1, c2 the positive acceleration constants; r1, r2 the random numbers, uniformly distributed within the interval [0, 1]; t the number of iterations so far; g the index of the best positioned particle among the entire swarm.
Flow chart for PSO:

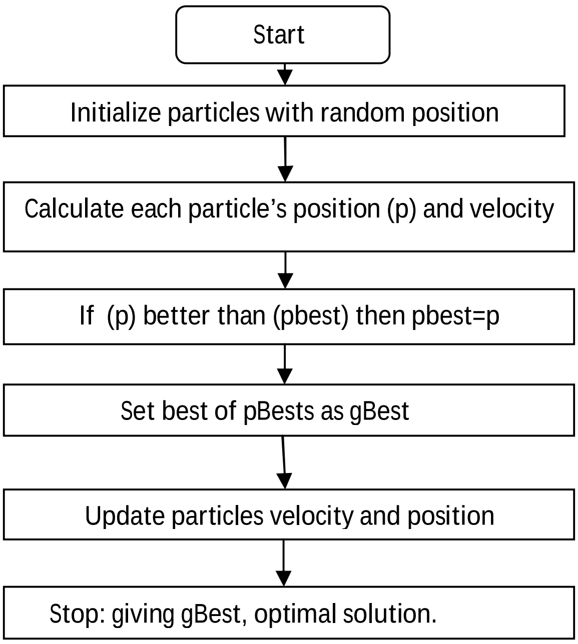

Fig 4:-Flow chart for PSO

Pi the position of pbest for the particle i; and Pg is the position of gbest for the entire swarm. In (1), c1 and c2 are used to scale the contribution of the cognitive and social components respectively. The random numbers r1 and r2 introduce a stochastic element to the algorithm. In the particle swarm optimization process, the velocity of each particle is adjusted according to (1) so that the particle oscillates around the locations of Pi and Pg. c1 and c2 are often set to 2.

The inertia weight w is employed to control the impact of the previous history of velocities on the current velocity, influencing the trade-off between global and local exploration abilities of the flying points. A larger inertia weight provides global exploration while as the smaller inertia weights tend to provide local exploration to fine-tune the current search area. Suitable selection of inertia weights provides a balance between global and local exploration abilities and so it needs only little iteration.

If we choose the population M = 9, then the PSO-based method could not compete with the four-step search method computationally unless Itermax <=5.Obviously, some blocks may require more than four iterations to find a good match, whereas for other blocks sufficiently good matches might already be found in the first several iterations. in order to achieve both low computational complexity and high estimation accuracy. We initialize the particles with random positions. After initializing the particles we have to find the position of each particle and their velocity. After initialization is done then we have to find the pbest value and the gbest value. Find the pbest value and the gbest value for the 9 particles. The location of the overall best value, obtained so far by any particle in the population, is called gbest .After getting the best value we will update the velocity and position for each particle.

Therefore, the following stopping criteria: When the total number of iterations reaches a pre-determined maximum number of iterations allowed (Itermax), the search is stopped. We selected the maximum iteration number Itermax to be 5.

V. RESULTS AND DISCUSSION

A. PSNR values:
The comparison of PSNR values for various video sequence for the full search method and the diamond search method and PSO method. Here while analyzing the full search method and the diamond search method and PSO method it could be seen that the PSNR value is higher for PSO search method when compared to the full search method and diamond search method.

TABLE 1 PSNR VALUES

Sequence	FS	DS	PSO
Stefan	24.12	22.85	26.39
Foreman	33.91	33.72	36.50
Car phone	33.49	32.49	37.28

B. Number of computations:

The average number of candidate blocks searched per current block. The comparison of number of computations done for various video sequences for the full search method and the diamond search method and PSO method. Here while analyzing the full search method and the diamond search method and PSO method it could be seen that the number of computations for full search method is higher when comparing to diamond search method and PSO method. The number of computations for PSO method is lower when comparing to other methods.

TABLE 2
AVERAGE NUMBER OF SEARCH BLOCKS

Sequence	FS	DS	PSO
Stefan	201	18.84	9.04
Foreman	201	16.31	8.72
Carphone	201	16.94	6.49

VI. CONCLUSION

In this paper, Particle Swarm Optimization was introduced to reduce the search points without the degradation of the image quality. Block matching method for motion estimation based on a particle swarm optimization provides accurate motion estimation with very low complexity in the context of Motion estimation. The PSO approach is capable of achieving high accuracy in block matching.

REFERENCES

[1] Zaynab Ahmed , Abir Jaafar Hussain and Dhiya Al-Jumeily, “ Fast Computations of Full Search Block Matching Motion Estimation”, ISBN: 978-1-902560-25-0 © PGNet(2011).

[2] Xuan Jing and Lap-Pui Chau , “An Efficient Three-Step Search Algorithm for Block Motion Estimation”, IEEE Transactions on Multimedia(2004).

[3] S. Zhu, K.-K. Ma, “A new diamond search algorithm for fast block-matching motion estimation”, IEEE Transactions on Image Processing(2000).

[4] H. Jia, L. Zhang, “A new cross diamond search algorithm for block motion estimation”, in Proceedings of IEEE International Conference on Acoustics,Speech, and Signal Processing(2004).

[5] 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).

[6] Ping Zhang,Ping,Wei,Hongyang Yu and Zhengning Wang, “Simplex Particle Swarm Optimization For Block Matching Algorithm”, International Symposium on Intelligent Signal Processing and Communication Systems(2010).

[7] S. Adedoyin,“An evolutionary strategy based motion estimation algorithm for H.264 video codecs”, IEEE Transactions on Image Processing(2005).

[8] Jing Cai 1, W. David Pan, “On fast and accurate block-based motion estimation algorithms using particle swarm optimization”, doi:10.1016Elsevier(2010).

[9] R.C. Eberhart, Y. Shi, “Comparison between genetic algorithm and particle swarm optimization”, in: Proceedings of the 7th International Conference on Evolutionary Programming(1998).