Middle-East Journal of Scientific Research 23 (12): 2905-2912, 2015
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2015.23.12.22774

A Novel Hybrid Differential Evolution Adaptive Approach for Block Matching In Motion Estimation

[1]K. Priyadarshini and [2]D. Jackuline Moni

[1]Department of ECE, Trichy Engineering College, Trichy, Tamilnadu, India
[2]Department of ECE, Karunya University, Coimbatore, Tamilnadu, India

Abstract: Motion estimation performance is an important concept in video coding applications. Among various motion estimation approaches Block-matching (BM) algorithms are the best methods due to their simplified and effective ways of hardware and software implementation. Block matching motion estimation is a popular method in developing video coding applications. Differential Evolution (DE) is one of the most prominent new evolutionary algorithms for solving real-valued optimization problems. In this paper, new algorithm has been proposed for reducing the number of search points using a hybrid differential evolution adaptive approach for motion estimation. The conventional differential evolution has been modified to provide accurate solutions in terms of Peak Signal To Noise Ratio (PSNR), Search Points (SP), Motion Estimation processing time. The population based collective learning process, self adaptation and robustness are some of the key featuers of evolutionary algorithms when compared with other global optimization techniques. Here we propose an optimized algorithm method which uses a population management mechanism, adaptive approach for cross over rate and local search routine to improve convergence. The results of proposed method are compared with those of DE and other motion estimation algorithms. The limitations such as computational time, search parameter, initial search and search space are overcome in the proposed method. The proposed technique saves computational time up to 94% and 47% improvement of PSNR when compared with other published methods.

Key words: Motion Estimation (ME) · Differential Evolution (DE) · Hybrid Differential Evolution (HDE) · Search Points (SP) · Peak Signal To Noise Ratio(PSNR)

INTRODUCTION

The development of new advanced video coding standards especially H.264 have introduced new challenges during the motion estimation (ME) process in interframe prediction [1]. The prediction of the motion vector is a major and important problem in the process of motion estimation since it can achieve significant compression ratio by exploiting the temporal redundancy existing in a video sequence. Inter-prediction motion estimation is a common tool used in video coding standards. To achieve best compression performance the video coding standards like H.264/MPEG-4 employ hybrid approach [2]. To target the real time processing in emerging video coding applications many efficient algorithms are designed and implemented. With the aim of complexity reduction in the process of video coding, analysis of several ME methods like Block Matching (BM) algorithms, optical flow method, parametric-based models and pel-recursive techniques are carried out [3]. Considering the factors of accuracy, simplicity and effectiveness BM is the most popular technique. In BM algorithms, a video current frame is partioned into non-overlapping blocks of equal size and the best matched block is determined using matching criteria within a predefined search window [4]. Among the various Block Matching algorithms, Full Search (FS) algorithm is the best optimal method since it can determine the best matching block in the search area and this algorithm requires maximum number of computations. In order to reduce the computation complexity many fast and efficient motion estimation algorithms are proposed [5]. In this research work, we propose a novel fast and efficient block motion estimation based on Hybrid Differential Evolution with initialization of adaptive parameters during the search process.

Corresponding Author: K. Priyadarshini, Department of ECE, Trichy Engineering College, Trichy, Tamilnadu, India.

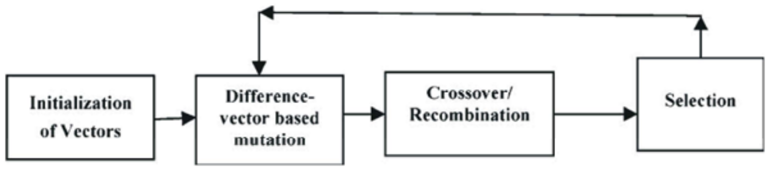

Fig. 1: Steps followed in Differential Evolution

To focus the problem of local optima in motion estimation various algorithms like Genetic Algorithm (GA) [6], Particle Swarm Optimisation (PSO) [7], Differential Evolution (DE) [8] are in research towards the searching methods. One of the most powerful algorithms of evolutionary computation is Differential evolution due to its excellent convergence characteristics and a few control parameters like fitness function, cross over ratio.

Basic Differential Evolution (De) Model: Differential evolution is an optimization algorithm developed by Price and Storn, [9] which solves real valued problems based on the principal of natural evolution. DE is a global optimization technique based on population which is simple to implement, reliable and fast. Fig. 1 shows the steps followed in Differential evolution. The performance of DE is mainly dependent on three operators mutation, recombination and selection process. It is an iterative process and the performance of DE algorithm is sensitive to the values of the parameters. r1, r2, r3 are random nos. generated within the interval [1, N]. "F" variation factor is a real number of the interval [0, 2]. According to the cross over strategy, the old and new individual exchange part of the code to form a new individual. "CR" is the cross over probability in the interval [0, 1]. Differential evolution allows several stratagies for optimization represented as DE/x/y/z, where x denotes to the vector used to generate mutant vectors, y denotes the number of difference used in the mutation process and z is the cross over scheme in the cross over operation. Evaluation is done with the parameter setting for DE on the Rastrigin's functions. Their experimental results revealed that the global optimum searching capability and the convergence speed are very sensitive to the choice of control parameters *NP, F* and *Cr* [10].

The DE/rand/1/bin policy options as mutation operation, crossover operation, selection operation are followed till termination criterion is met to produce a new generation of individual.

Differential Evolution for Motion Estimation: Motion estimation is a multi-step process which proceeds as Motion Vector prediction, calculation of search range and determination of the optimization criterion. Fig. 2 depicts the concept of Macroblock in motion estimation process. The steps followed in the Differential Evolution algorithm is as follows.

Step 1: The population Initialization ' G'

$$x_i^{(G)} = \{x_{i,1}^{(G)}, x_{i,2}^{(G)}, ..., x_{i,D}^{(G)}\}, i = 1,2,..., NP. \quad (1)$$

The population size NP is an algorithm control parameter assigned and remains constant throughout the optimization process. Here, in the Motion estimation process each individual represents a search pixel location in the macroblock.

Step 2: Mutation

During one generation for each vector x (*G*)*i*, DE employs mutation and crossover operations to produce a trial vector:

$$u_i^{(G)} = \{u_{i,1}^{(G)}, u_{i,2}^{(G)}, ..., u_{i,D}^{(G)}\}, i = 1,2,..., NP. \quad (2)$$

Then a selection operation is used to choose vectors for the next generation (*G*+1).The initial population is usually selected uniformly randomly between the lower *xj*, *low* and upper *xj*, *upp* bounds defined for each variable *xj*. These bounds are specified according to the block matching process technique applied to the video sequence.

Step 3: Mutation Operation: Mutation of each population vector $x_i^{(G)}$ produces a mutant vector $v_i^{(G)}$.

$$v_i^{(G)} = \{v_{i,1}^{(G)}, v_{i,2}^{(G)}, ..., v_{i,D}^{(G)}\}, i = 1,2,..., NP. \quad (3)$$

Step3: Cross over Operation.

After mutation, a 'binary' crossover operation forms the trial vector u (*G*)*i* according to the target vector x (*G*) *i* and its corresponding mutant vector v (*G*)*i*.

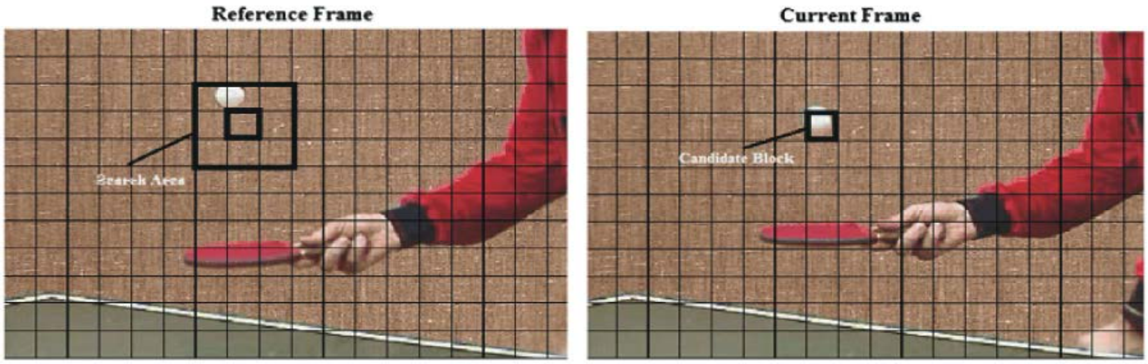

Fig. 2: Macroblock in motion estimation process

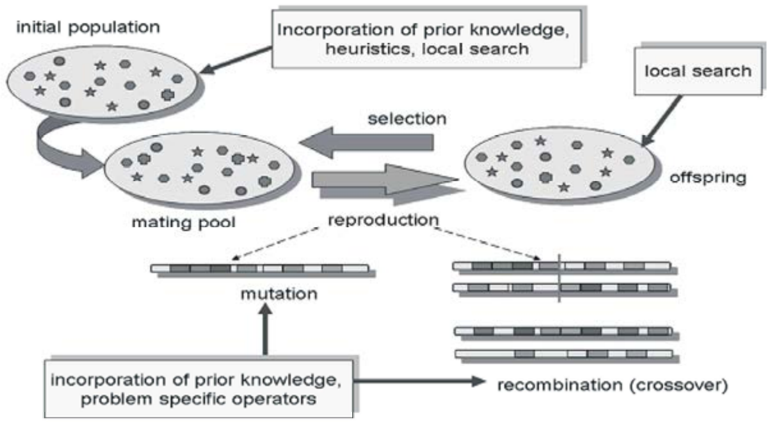

Fig. 3: Hybridization concept in evolutionary process

$$u_{i,j}^{(G)} = \begin{cases} v_{i,j}^{(G)} & \text{if } rand\ (0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j}^{(G)} & \text{otherwise} \end{cases}$$

$$i = 1,2,...NP \text{ and } j = 1,2,...,D. \quad (4)$$

CR is a crossover control parameter or factor within the range [0, 1) and presents the probability of creating parameters for a trial vector from the mutant vector. Index *jrand* is a randomly chosen integer within the range [1, *NP*]. It ensures that the trial vector contains at least one parameter from the mutant vector. Here we have described the binary crossover operation (*bin*). The other DE crossover operation that could also be used in optimizations is exponential (*exp*).

Step 4: Selection Operation

The DE algorithm uses a greedy selection. The selection operator selects between the target and corresponding trial vectors. A member of the next generation becomes the fittest vector, i.e. vector with the better fitness value. During the optimization process the quality of the motion vector is evaluated using the fitness function, Mean Absolute Difference (MAD) corresponding to each Motion vector.

The control parameters of differential evolution are assumed to be F, Cr & Np Fig. 3. represents the Hybridization concept in an evolutionary process. Also researchers attempted to embed different local search techniques in basic DE, to improve its exploitation abilities [10]. In this section, the hybrid DE Local search algorithms primarily explore a small neighborhood of a candidate solution in the search space until a locally optimal point is found or a time bound is elapsed algorithms. Typically in LS, every candidate solution has more than one neighbour solution; the choice of which one to move to is taken using only information about the solutions in the neighbourhood of the current one, hence the name local search.

Proposed Hybrid Differential Evolution with Adaptive Cross over

Operator for Motion Estimation: The proposed algorithm solves Block Matching process as an optimization problem. The important features of this algorithm is that it uses a population management mechanism to improve exploration, an adaptive mechanism for cross over rate and a local search routine to improve convergence. Initialization of population to the generation of off-springs is incorporated with the concept of motion estimation algorithms. Population is initialized by local search technique within the initialized population member and among the off-springs.

The algorithm is summarized as follows:

Step 1: Specify the DE parameters.
Step 2: Select the initial population.
Step 3: Evaluation of population using the fitness function.
Step 4: Create off-springs and evaluate their fitness.
Step 5: Compare the fitness of off-spring with the parents
Step 6: Check the size of updated new population with existing population using eqn. (2) and (3).
Step 7: Stop the iteration process if size of updated new population is maximum which is the termination strategy. Otherwise go to step 3.
Step 8: The termination point determines the corresponding Motion vector.

Fig. 4 shows the flow in which the fitness value is calculated by using the objective function. The proposed algorithm uses a fitness calculation strategy to reduce the evaluation of MAD values and it requires only minimum search points.

Fitness Function: The fitness function of the proposed algorithm is defined as MAD between the pixel values in current block of the reference frame and with the location of the previous frame. The parameters which affects convergence speed of the algorithm are the Scale factor (F) and Cross over factor (CR). Number of iterations, determine the tradeoff between accuracy and computational complexity. The parameter *CR* controls how many parameters in expectation are changed in a population member. For low value of *Cr*, a small number of parameters are changed in each generation and the stepwise movement tends to be orthogonal to the current coordinate axes. On the other hand, high values of *Cr* (near 1) cause most of the directions of the mutant vector to be inherited prohibiting the generation of axis orthogonal steps. Hybrid Differential Evolution parameters are assigned as follows:

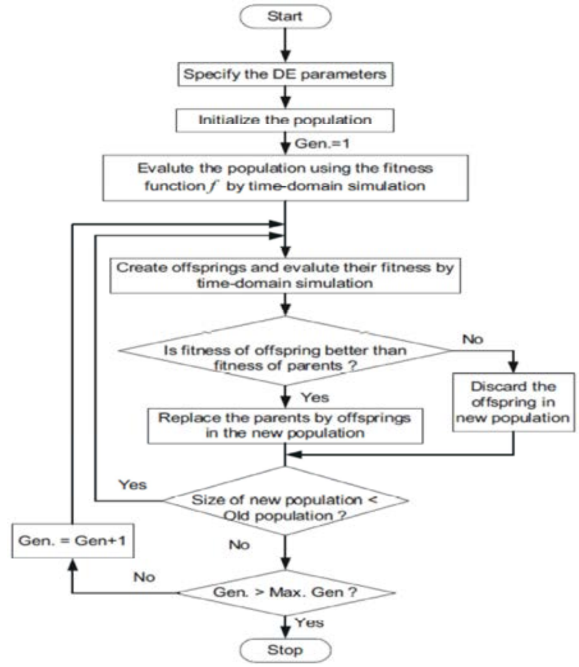

Fig. 4: Flow chart of the proposed algorithm

Scaling factor (F)= Triangular Distribution;
No. of decision variables = 06;
Cost Function = RASTRIGIN; Maximum iteration = 5

Performance Analysis: To compare the performance of the proposed algorithm approach, various search algorithms such as FSA, TSS, FSS, DS, DE have been implemented in our simulations. To perform the experiment 30 frames are selected from the test video sequences assigned. For comparison purposes, all nine video sequences in Fig. 5 have been used. During the comparison process, three relevant performance indexes have been considered: the distortion performance, search points efficiency and the motion estimation processing time.

Distortion Performance: The first step is the comparison of their distortion performance, which is by Peak-Signal-To-Noise Ratio (PSNR) value. The PSNR is given by eqn., where MSE is the mean square between the original frames and those compensated by the motion vectors.

$$MAD = \frac{1}{N^2} | \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} | C_{ij} - R_{ij} |$$

$$MSE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (5)$$

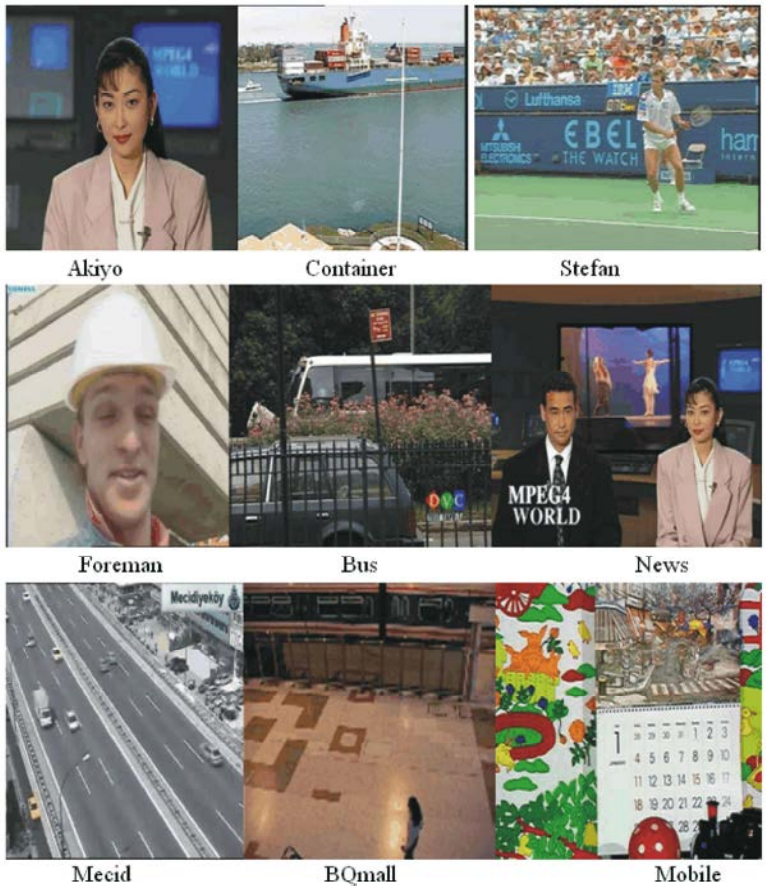

Fig. 5: Test Video Sequences

Table 1: Test sequences used in comparison process

Video Sequence	Format	Frame Size	Motion Type
Akiyo	QCIF	352X288	Medium
Container	QCIF	352X288	Low
Stefan	CIF	352X288	High
Foreman	QCIF	352X288	Medium
Bus	CIF	352X288	High
News	QCIF	352X288	Medium
Mecid	QCIF	320X240	Medium
BQmall	CIF	160X120	High
Mobile	CIF	352X288	Medium

where N is the side of the macro bock, Cij and Rij are the pixels being compared in current macro block and reference macro block, respectively.

$$PSNR = 10Log_{10}\left[\frac{(Peak\ to\ peak\ value\ of\ original\ data)^2}{MSE}\right] \tag{6}$$

In Table 2, PSNR values are compared with the proposed algorithm. In the proposed motion estimation each frame is divided into N X N non-overlapped blocks and of each block a search is made between the reference frame and the current frame over a random area of the image. The main aim for the search is to determine the position that minimizes a distortion measured between the two sets of pixels comprising the blocks. The relative displacement between the two blocks is the motion vector. MSE is the mean square deviation between reconstructive frame and original frame which is represented by eqn.5. To evaluate the computational complexity average searching points are used. The search efficiency is used to calculate the computational complexity of the algorithm used. The efficiency of search points is determined by calculating the average number of search points for motion estimation.

Simulation Results: In the simulations carried out, first 30 frames of the standard test video sequence listed in Table 2. The macroblock size for motion estimation is considered as 16*16 and analysed. The analysis of the proposed algorithm is compared with other algorithms and analysis is carried out. Table 2 denotes the results of the average PSNR values of various algorithms.

Table 2: Comparison of Average PSNR results in db of different methods

Video Sequence	FS	TSS	FSS	DS	DE	HY-DE
Akiyo	54.66	42.61	42.62	42.63	53.40	54.40
Container	54.75	40.75	40.75	40.75	53.69	54.69
Stefan	45.75	35.09	35.00	35.00	43.72	44.72
Foreman	49.95	39.47	39.52	39.64	46.26	47.26
Bus	45.09	33.25	33.01	33.17	41.08	42.08
News	52.39	40.83	40.82	40.84	50.86	51.86
Mecid	48.07	37.43	37.27	37.14	46.93	47.93
BQmall	51.75	40.71	40.60	40.66	48.38	49.38
Mobile	43.06	34.38	34.52	34.61	41.01	42.01

Table 3: Comparison of Average number of search points during motion estimation

Video Sequence	FS	TSS	FSS	DS	DE	HY-DE
Akiyo	231.95	22.80	15.67	6.38	4.46	4.03
Container	229.69	22.80	15.66	6.43	4.53	4.09
Stefan	229.79	22.94	18.53	9.28	7.89	7.11
Foreman	227.46	22.96	18.73	9.09	8.00	7.20
Bus	224.64	23.47	20.66	11.02	9.37	8.46
News	230.54	22.79	15.76	6.51	4.72	4.25
Mecid	226.85	22.81	16.16	6.80	5.09	4.58
BQmall	231.13	22.81	15.88	6.59	4.76	4.29
Mobile	229.78	22.84	17.00	7.48	6.72	6.08

Table 4: Comparison of Computation time in seconds for Motion Estimation

Video Sequence	FS	TSS	FSS	DS	DE	HY-DE
Akiyo	255.24	4.18	3.20	3.20	0.21	0.20
Container	357.6	3.08	2.01	2.04	0.14	0.13
Stefan	238.8	2.57	3.48	3.00	0.20	0.19
Foreman	397.2	3.80	2.06	2.75	0.23	0.21
Bus	225.6	2.56	4.25	3.92	0.23	0.23
News	244.8	2.26	3.16	2.12	0.18	0.19
Mecid	232.2	2.82	2.29	3.11	0.13	0.13
BQmall	542.4	3.75	2.88	3.38	0.19	0.19
Mobile	542.4	2.92	3.03	4.10	0.32	0.32

The PSNR value of the video sequences which have violent motion like Stefan, Bus have minimum PSNR value in the FS method when compared with other video sequences. The proposed Hybrid-Differential algorithm performs better than all other proposed algorithms. Table 3 denotes the results of number of search points and minimum search points for akiyo and container sequence is achieved using the proposed algorithm. That is one fifty that of FS for akiyo, Container, News, MECID, Bqmall sequence and one fifth for TSS and one fourth times that of FSS. The scale factor ◎◎ and cross-factor CR have a great impact on the performance of the algorithm, such as the quality of the optimal value and convergence rate.

In order to compare the performance of various algorithms, each algorithm is executed with 30 frames for different video sequences and the parameters PSNR, search points and computational time are determined. Table 2, Table 3 and Table 4 shows the values predicted during the experimental process.

Fig. 6 shows the frame by frame PSNR comparison between different BMA'S on mecid and Stefan test sequences for the first 30 frames. Fig. 7 shows the Motion vector distribution for mecid and Stefan sequences and Fig. 8. depicts the frames during the motion estimation process. The simulated figures shows the improvement in PSNR value and reduction in search points with maintenance of the image quality.

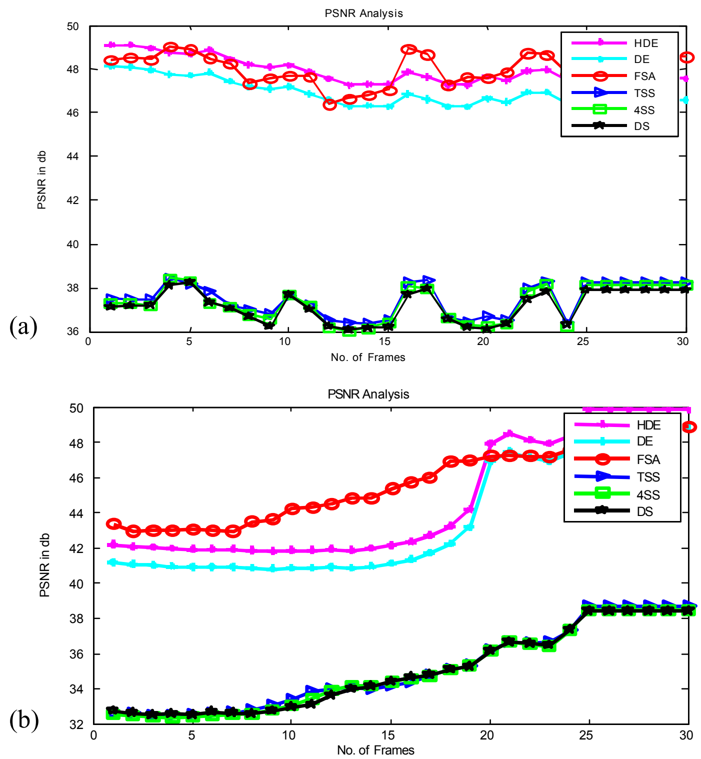

Fig. 6: Frame-wise PSNR comparison between different BMA'S on test sequences (a) mecid and (b) Stefan considering 30 frames

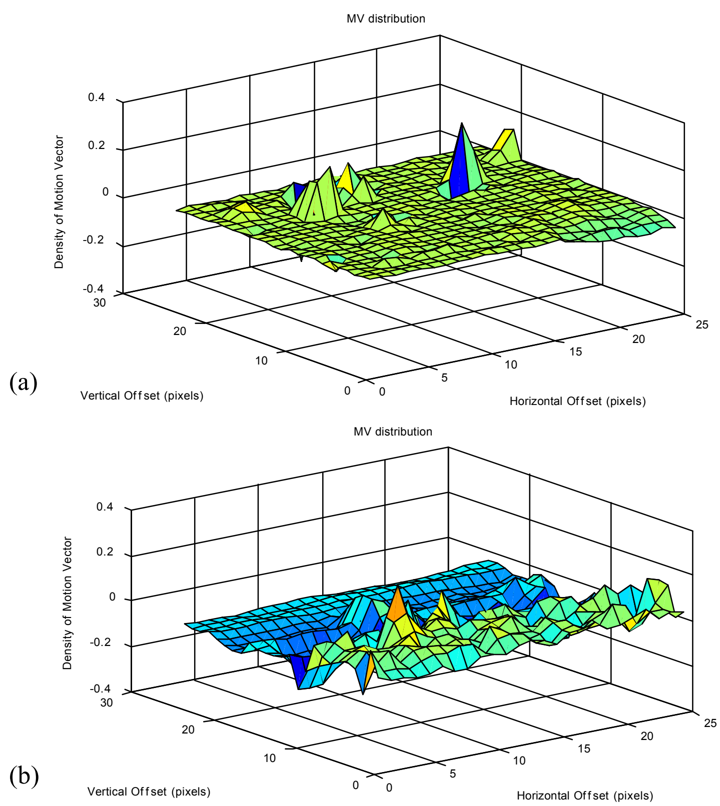

Fig. 7: Motion vector distribution for a) Mecid sequence b) Stefan sequence

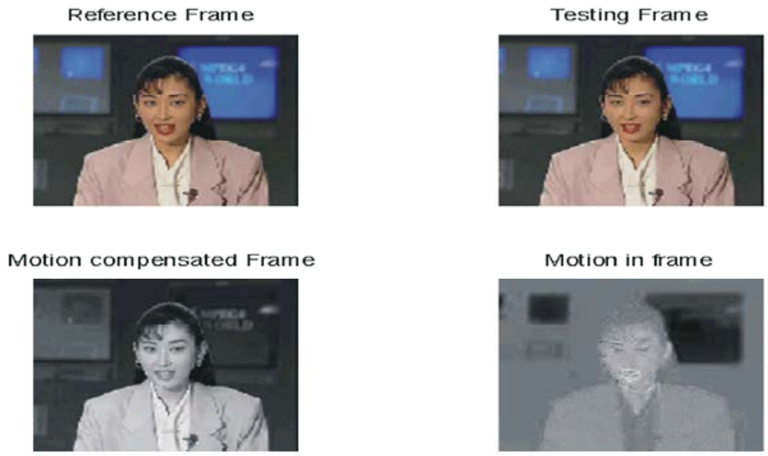

Fig. 8: Motion Estimation on test video Akiyo

CONCLUSION

In this paper, a novel algorithm based on adaptive Hybrid Differential Evolution has been proposed to improve PSNR and reduce search points in the motion estimation procedure. The conventional DE has been modified with hybrid differential evolution algorithm with a local search (LS) technique and an adaptive mechanism for crossover operator to solve a set of single objective optimization in the method of motion estimation process. The proposed technique achieves accurate motion estimation, 94% improvement of average number of search points with low computational complexity and 47% high accuracy in terms of PSNR.

REFERENCES

1. Sullivan, G.J., J.R. Ohm, W.J. Han and T. Wiegand, 2012. "Overview o f the High Efficiency Video Coding (HEVC) Standard"; Circuits and Systems for Video Technology, IEEE Transactions on, 22(12): 1649-1668.
2. Kim, D.H., A. Abraham and J.H. Cho, 2007. A hybrid genetic algorithm and bacterial foraging approach for global optimization, Info. Sci., 177: 3918-3937.
3. Huang, T., C. Chen, C. Tsai, C. Shen and L. Chen, 2006. Survey on Block Matching Motion Estimation Algorithms and Architectures with New Results. Journal of VLSI Signal Processing, 42: 297-320.
4. Li, S., W. Xu, N. Zheng and H. Wang, 2000. A novel fast motion estimation method based on genetic algorithm, Acta Electronica Sinica, 28: 114-117.
5. Yoel Tenne, 2012. A computational intelligence algorithm for expensive engineering optimization problems, Engineering Applications of Artificial Intelligence, 25(5): 1009-1021.
6. Eberhart, R.C. and Y. Shi, 1998. Comparison between genetic algorithm and particle swarm optimization, in IEEE International Conference Computation, Anchorage, AK, May, pp: 611-616.
7. Storn, R. and K. Price, 1997. “Differential evolution-a simple and efficient heuristic for global optimization over ntinuous spaces, ” Journal of Global Optimization, 11(4): 341-359.
8. Liu, J. and J. Lampinen, 2005. “A fuzzy adaptive differential evolution algorithm,” Soft Computing, 9(6): 448-462.
9. Zhang, W.G., H.P. Chen, D. Lu and H. Shao, 2008. “A novel differential evolution algorithm for a single batch-processing machine with non-identical job sizes, ” in Proceedings of the 4th International Conference on Natural Computation (ICNC’ 08), pp: 447-451.
10. An Improved Differential Evolution Algorithm Based on Adaptive Parameter Zhehuang Hindawi Publishing Corporation Journal of Control Science and Engineering, 2013, pp: 1-6.
11. Storn, R. and K. Price, 2003. “Differential evolution formulti-objective optimization,” Evolutionary Computation, 4: 8-12.