International Journal of Computer and Electrical Engineering, Vol. 2, No. 4, August, 2010
1793-8163

Probability Based Search Motion Estimation Algorithm Using Mean Correction

Deepak J. Jayaswal [1] and Mukesh A. Zaveri [2]

Abstract—**We propose a stochastic approach to determine the motion vector (MV) for block matching algorithm (BMA). This approach allows us to exploit random distribution of motion vector in successive video frames from which the initial candidate predictors are derived. The derived predictors are the most probable points in search window, which will assure that, the motion vectors in the vicinity of center point and at the edge of the search window does not miss out, as it does for earlier algorithms like Three step search(TSS), Four step search(FSS), Diamond(DS), etc and refinement stage used in the algorithm will allow us to extract true motion vector so that the picture quality is as good as Full search(FS) which is the optimal algorithm. The novelty of the proposed algorithm is that the search pattern derived is not static but can dynamically shrink or enlarge to account for small and large motion. It is important to note that for the first time mean correction technique is introduced in video codec to improve PSNR with early termination of the algorithm. The Simulation result shows that our proposed algorithm outperforms all sub-optimal algorithms in terms of quality and speed up performance and in many cases PSNR of proposed algorithm is comparable or better than Full Search.**

Index Terms— **Motion vector, Block matching algorithm, Three step search, Four step search, Diamond search, Full search**

I. Introduction

Motion estimation (ME) and compensation are the keys to high quality video coding [1]. ME is a process of estimating the motion of macroblock (MB) of some predefined size from the reference frame to generate the current frame. Block Matching based motion estimation is used in most video codecs, including MPEG-2, MPEG4 and H.263 [2]. ME is also a key component in the digital restoration of achieved video and for post production and special effect in the movie industry. ME and the exploitation of the strong correlation among the successive frames allows us to encode and transmit the motion vectors along with the error frame obtained using the regenerated current frame and the actual current frame and hence, reduces the number of bits used to convey the information. To achieve this bit reduction, various approaches and algorithms have been proposed in the literature [1] [2] [3] [24] [27]. The most accurate BMA is the exhaustive FS method, which exhaustively evaluates all possible macro blocks (p x p) over a predetermined search window of size (2p+1) x (2p+1) to find the best match. The estimated motion vector is the best match achieved for a predefined block distortion measure (BDM). The only disadvantage of this method and perhaps the biggest flaw is the high computational cost associated with it. Other algorithms with reduced number of computations, for example, the successive elimination algorithm (SEA) [4], a new three step search [5], a novel four-step search (FSS) [6], efficient four step search [7], unrestricted center biased diamond search (UCBDS) [8], cross search [9], fast full search motion estimation [10], complexity bounded motion estimation [11], new fast algorithm for estimation of block motion vector [12], dynamic search window algorithm [13], predictive coding based on interframe efficient motion estimation [14], displacement measurement and its application in image coding [15], a new efficient block-matching algorithm for motion estimation [16], fast variable block-size motion estimation algorithms based on merge and split procedure [17], Among these algorithms, the SEA is similar to the full search method except, the first one eliminates certain search points based on the Minkowiski's inequality. Further reduction in number of search points was achieved in TSS algorithm which starts with a step having nine uniformly spaced search points which get closer after every step until the step size reduces to 1. The best candidate search point in the previous step becomes the center of the current step. The main drawback of TSS is the relatively large search pattern in the first step having a distance of 4, which renders it inefficient for finding blocks with small motions. In order to exploit the characteristics of the center-biased motion vector distribution, FSS algorithm was proposed to speed up the search mechanism. It utilizes a nine-point search pattern on a 5 X 5 grid in the first step. FSS requires only four search steps as it starts with a smaller search grid pattern. The total number of candidate search points in FSS actually ranges from 17 (the best case) to 27 points (the worst case). The main drawback of FSS is the relatively small search pattern in the first step having a distance of 2, which renders it inefficient for finding blocks with large motions. TSS, FSS algorithm are static that is they have fixed search pattern and restricted window size, and are inefficient for predicting true motion vector for low or high motion. Other algorithm like appeared in [18–27] uses motion vector prediction (MVP) technique and has shown significant performance improvement in terms of quality. These algorithms provide better peak to signal ratio (PSNR) as compared to TSS, FSS, DS, etc. and are not static but are very complex and needs memory to store motion vectors for

Manuscript received September 2, 2009.

D. J. Jayaswal is with the St Francis Institute of Technology, Mumbai, India. (Phone: +91-22-28928585; fax: +91-22-28954787; e-mail: djjayaswal_vcet@yahoo.com).

M. A. Zaveri is with the Sardar Vallabhai National Institute of Technology, Surat, India (e-mail: mazeveri@coed.svnit.ac.in).

prediction. This makes it difficult for multimedia application where porting of video codec for embedded processor is required as well as for video streaming application for video on demand. Almost all of the BMAs make explicit and implicit assumptions that the matching distortion increase monotonically as the checking point moves away from the global minimum or the error surface is unimodal over the global window. Indeed, this assumption is not always true. Consequently, the resultant MVs may be trapped in a local minimum. Most BMAs exhibit proper behavior provided the following prerequisites are met [17]: 1) object displacement is constant within a block of pixels; 2) pixel illumination between successive frame is spatially and temporally uniform (this constraint can be relaxed in the BMA with luminance correction); 3) motion is restricted to translation; 4) matching distortion increases monotonically as the displaced candidate block moves away from the direction of the exact minimum distortion. Most of these conditions are usually not met for real-life video sequences, but a large number of ME algorithms still perform reasonably well. There are two major disadvantages of these algorithms (i) these algorithms yield different performance on different video sequences and (ii) their high complexity. In this paper we suggest a stochastic approach to determine the initial candidate predictor that is the most probable search point for first iteration and it is followed by refinement stage which allow us to extract true motion vector so that picture quality is as good as FS. The designed search pattern as mentioned earlier can dynamically shrink or enlarge depending on mean and threshold criteria to account for small and large motion. The algorithm proposed has a the search pattern which is derived from the knowledge of probability distribution of motion vector in given search window and concept of mean correction, will ensures that, 1) the search is not trapped in the local minima, 2) the search does not miss out on the motion vector found in the central region, 3) the search is terminated early by using mean correction and threshold without sacrificing PSNR, 4) PSNR is improved using mean correction technique, 5) computations are less 6) complexity is low, 7) memory is not needed to store motion vectors as it is required in most predictive ME algorithm. This paper proposes a PBSMC algorithm and the organization of the paper is as follows: Section II presents the review of motion estimation algorithm. Section III presents performance evaluation measurements. Section IV presents proposed algorithm and explains the algorithm in detail considering the steps involved. Section V presents some simulation results of our proposed algorithm in comparison with the algorithms like the FS, SEA, TSS, 4SS and DS.

II. Review Of Motion Estimation Algorithm

Suboptimal motion estimation algorithm develop in recent years can be broadly classified as follows.

1) *Heuristic Search Techniques:* Instead of searching all candidates within search area, it looks for less number of candidate MVs in search area. The choice of the position is driven by some heuristic criterion in order to find the absolute minimum of cost function. The most famous variant of this technique are TSS, NTSS, FSS, and DS.
2) *Fast Exhaustive Search Technique:* In these technique reduction of search points is obtained by removing many of the nonoptimal search position without losing the optimality of the FS algorithm [SEA].
3) *Hierarchical or Multiresolution Technique:* The MVs are searched for a low resolution image and the refined in the normal resolution.
4) *Spatio-Temporal Correlation Technique:* The MVs are selected using the vectors that have already been calculated in the current and in the previous frames. Often this vector is starting point of a refinement stage

The algorithm presented in this paper belongs to heuristic and spatio-temporal technique category: the algorithm in facts finds the best initial predictor MV between a restricted set of vectors and refines it in a successive phase. Due to the presence of initial predictor which is decided on knowledge of MV distribution in given search window and its ability to switch between search patterns and improve quality by mean correction the algorithm is named as probability based search motion estimation with mean correction(PBSMC).

III. Performance Evaluation Measurement

There are several performance measure function to test the efficiency of a MV, each of which weighs the pointto-point difference between two MBs: the PBSMC uses the SSE function (sum of square error) as it offers a good evaluation precision.

$$SSE(m,n) = \sum_{i,j} [B(i,j) - B(i+m, j+n0]^2$$

with i,j Є macroblock

where B(i,j)is the pixel value in the (i,j) position of the current frame and B'(i+m , j+n)is the corresponding pixel value of the previous frame, in a position that has been shifted by a MV of the (m,n) components.

IV. Proposed Pbsmc Algorithm

Our algorithm is based on stochastic approach which exploit random distribution of MVs in successive video frames for selection of search points within given search window, with the assumption that motion field varies slowly both spatially and temporally, therefore it is highly probable that MBs closer to the current MB, in time and space may show the same motion. If this assumption holds, instead of testing all possible MVs, as FS does, we can use the most probable MV that is the set of candidate predictors. Then the best candidate predictor, that is the MV with the lowest SSE, is sent to a refinement phase that allows obtaining the final MV. The algorithm described here operates in two phases. In first phase we determine the initial candidate predictor to begin the search and also ensure that search does not get trap in local minima and in second phase the refinement stage will look for true motion vector.

A. Initial candidate predictor:

The driving idea is to reduce the cardinality of the candidate predictor set as much as possible and, at the same time, to put the "best" candidate in the set that is those MV that describe at best the motion. In order to find the best

candidate for initial search,

We propose the basic search pattern as shown in Fig.1. The search points in the central diamond region, at location (0,0) (0,-1) (0,-2) (0,1) (0,2) (-1,-1) (-1,0) (-1,1) (-2,0) (1,1) (1,0) (1,1) (2,0) are 13 most probable points which are derived by determining the motion vector distribution probability over a search window of (2p+1)x(2p+1).

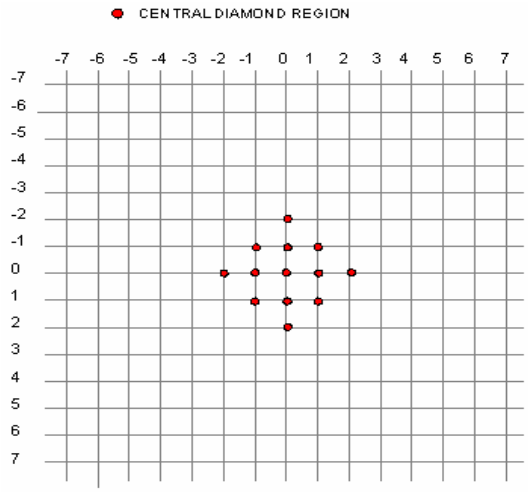

Fig.1. Basic search pattern of PBSMC

Table I depicts the percentage motion vector distribution probability over search window of 15x15 (p=7) for multiple frames of QCIF sequence using full search (FS) motion estimation algorithm. FS finds motion vectors by sequentially searching the whole 15x15 search window in the reference frame. A MB centered at each of the position in the window is compared to the MB in the target frame, pixel by pixel, and their respective sum of square (SSE) is then derived. The vector (i,j) that offers the least MAD is designated the motion vector MV(u,v) for the MB in the target frame. For each of the MV detected at location (i,j) in 15x15 search window a count is maintained for multiple frames of the sequence. The percentage MVP at a particular location (i,j) is

% MVP(i,j) = 100 * (Number of MVs found at location(i,j) / Total number of MVs)

= 100 * (Count(i,j) / Total number of MVs)

For example if we consider 75 frames with frame size of 320x240 and MB size of 8x8, we have total 40x30x75 numbers of MVs. If count at (i,j) location using FS equals

15 then **% MVP(i,j) = 15x 100 = 0.11%**

As observed from Table.1, the motion vector distribution is highly center biased with about 75% of the motion vectors being found in the central diamond region. To further justify our selection of 13 points in central diamond region, we analyze the shortcoming in earlier suboptimal algorithm like TSS, FSS, and DS. It is well known that the PSNR of these algorithms are low as compared to FS, the cause of failure of these algorithms is initial static search pattern which is unable to detect true motion vector that is TSS, FSS, and DS fails to find best match that FS has been able to find at particular location within 15x15 search window. To analyze the pitfalls of these algorithms we have generated error table using varieties of video sequences but we have shown result of akiyo sequence in Table. II - IV where each entry in table indicates the percentage error that is number of times TSS, FSS, and DS fail to find the true motion vector for multiple frames that FS has been able to find at particular location within 15x15 search window. The percentage error at location

(i,j) is

% error (i,j) = 100 * { 1- [(number of MVs found by TSS at (i,j) / number of MVs found by FS at (i,j)] }

The zero percent error at nine location in Table II-IV is reflection of initial search points of TSS, FSS, and DS respectively but error accumulation in inner diamond region at location (-1,0) (0,-1) (1,0) (0,1) is about 45% for TSS, 35% for FSS and 24% for DS, clearly diamond search performs better than other two algorithm but still fails to account for MVs in inner diamond region. Therefore it is mandatory for any motion estimation algorithm, not to miss out on any MVs found in central diamond region.

Based on the probability distribution of motion vector and error table, our next logical step was to decide on number of search points in proximity of central point that is mv(0,0), for this we selected maximum 13 most probable points in central diamond region. We found that this 13 most probable points as shown in Fig.1. gives better PSNR result than diamond search at cost of few additional points. However, from simulation result of Table VIII-IX, it is observed, this algorithm, just like diamond search, gets trapped in local minima and often fails to find motion vectors which are closer to edge of the search window. In order to solve the problem of local minima trap and also look for the motion vector located at edge of search window. We propose the search pattern of Fig.2 as initial search for first iteration with most probable points in central diamond region and next probable 8 points in outer diamond region located at (6,0) (-3,-3) (-3,3) (0,-6) (0,6) (3,-3) (3,3) (6,0) to take care of motion vector located at edge of search window.

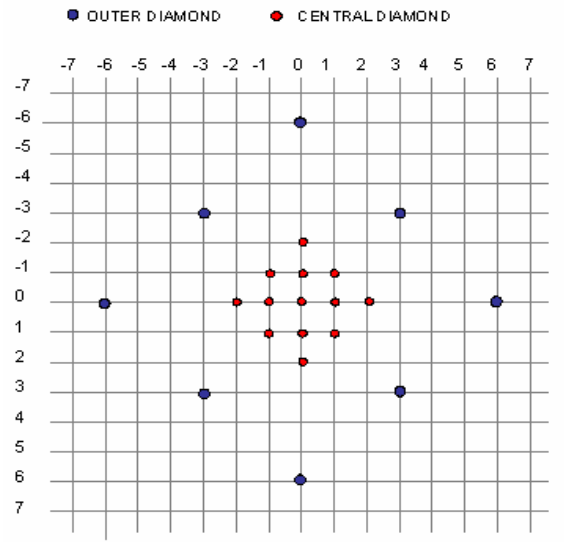

Fig.2. Search pattern of PBSMC (PBSMCT)

This pattern has ability to shrink or enlarge depending on criteria mentioned, the shrinking can be as small as five points and enlarge may be as big as twenty one points. After initial iteration, if minimum cost function is found in the central diamond region the refinement stage uses eight point search (EPS) described in section IV-B and if minimum cost

function is found in outer diamond the search algorithm uses DS to track true motion vector. Simulation result will show that this pattern do not get trap in local minima as well as look for true motion vector.

B. *Refinement Stage:*

In this phase the refinement vectors try to correct the best candidate predictor searched in first phase to achieve the true motion vector. The refinement is done on the vectors obtain in first iteration to ensure that algorithm does not get trapped in local minima. Refinement phase let the motion field evolve and provide the required resolution. Since there is not a preferred direction, the refinement vectors are placed as a grid, centered on the best predictor, made of four points on cross direction and four on diagonal directions for motion vector found in central diamond region. These correction vectors of the refinement grid have a very limited range, one pixel, but they are able to give a good correction in most cases. However, in some circumstances, for example the MV found at the edge of the search window, it is convenient to enlarge the grid to recover the correct motion, in other words use diamond search to locate true motion vector.

In our algorithm refinement grid shrinks if minimum SSE is found in central diamond that is EPS used. It enlarges if minimum SSE is found in outer diamond that is DS algorithm used. We consider the eight neighbors around the central point. If a match is found at the center, the search is terminated. If a match is found at any of the neighbors, it is considered as the central point for the next iteration in the search. The search then continues in this manner till a central point is found as the match.

- Eight point search (EPS)

The basic search-point configuration of the EPS is as shown in the Fig.3

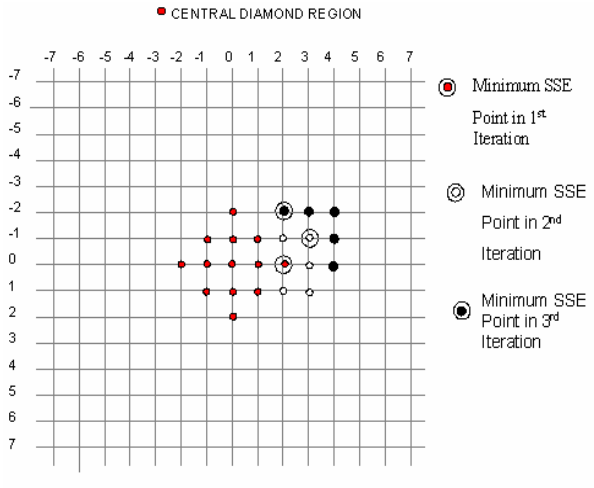

Fig.3. Eight point search pattern

C. *Quality Improvement using Mean Correction*:

In FS method the search is performed by considering all the position inside the reference frame within a search area around the current MB and testing all of them to find the position that minimizes a suitable cost function. The displacement between the position of the MB in the current frame and the position of the MB, in the reference frame, that minimizes the cost function becomes the motion vector of the MB. Although FS does find the MVs with minimum SSE but it does not assure minimum residual error, in many cases the minimum SSE as shown in Table VI at location (1,-4) is high, though this is minimum cost function but will lead to degraded PSNR.

The concept of mean correction introduce for the first time in this algorithm will always assure that the residual error is kept at minimum which not only improves PSNR but also helps for early termination of algorithm.

Concept of "mean correction"

Consider the following 8x8 macroblock.

33 34 33 32 31 32 33 34
33 32 33 33 31 33 32 33
34 33 32 31 33 34 33 33
32 32 34 32 33 34 33 34
34 33 32 33 33 31 33 32
31 32 33 33 33 33 33 34
33 32 33 33 33 33 34 34
33 33 33 33 33 34 34 34

(Macroblock A)

mean of this macroblock is 32.875 which can be rounded off to 33. Now, let us construct an artificial macroblock B with all the values being 33.

33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33
33 33 33 33 33 33 33 33

(Macroblock B)

Thus, the SSE is 46. Let us denote this by S1. Now, suppose that our algorithm has found a best match with an SSE, S2 = 125. Clearly, S1 is smaller than S2 and is therefore a better "match". Therefore, instead of sending the motion vector, we can directly send the mean of the current macroblock in such cases. At the receiver end, the macroblock will be reconstructed using this mean value. Therefore this process will assure better PSNR of the reconstructed macroblock using mean value. If S2 is smaller than S1, transmit the vector location of S2. It is important to note that the computation of mean itself takes some processing time. Mean computation requires 63 additions, 1 division and 1 rounding off operation. SSE requires 64 subtractions, 64 squaring operations and 63 additions. So a mean computation requires much less computation time than the time required for an SSE computation. Therefore, we have not accounted for it while calculating the search points.

D. *Early Termination of Algorithm:*

To reduce computation in first iteration and iteration to follow without sacrificing the quality of video the algorithm uses technique of early termination using fixed threshold.

In all of the video sequences that we analyzed using the FS, the average PSNR, was in the range of 22 dB to 45 dB. For early termination of search, selecting PSNR of 22dB will degrade the quality and choosing PSNR much above 45 dB increases the number of search points. Considering this fact,

we can safely assume that for all practical purposes, a PSNR of 45 can be considered to be a "good PSNR".

$PSNR = 10 * \log (255^2 / MSE)$

Therefore, $45 = 10 * \log (255^2 / MSE)$

Therefore, MSE = 2.056

Therefore, for an 8x8 macroblock,

SSE = 64 * 2.056 = 131 (approx.)

Thus, we can say that a match is a good match if the SSE is less than or equal to 131, the threshold SSE. The graph of Fig.4 justifies our selection of fixed threshold (45 dB) and effect of selecting lower and higher threshold on picture quality and number of search points.

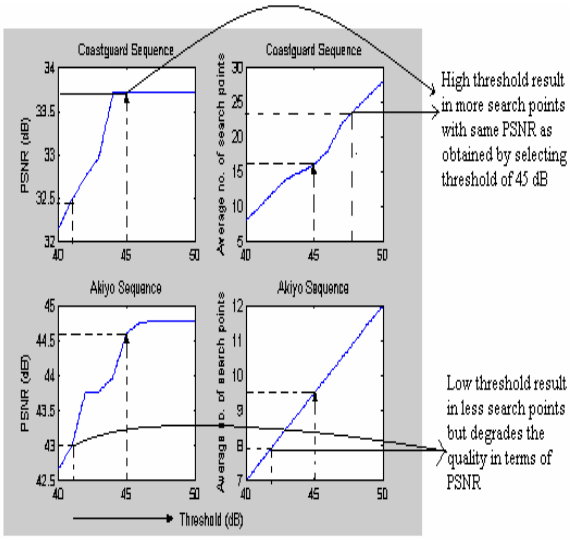

Fig.4 Effect of choice of threshold on picture quality and no. of search points

E. The PBSMC (PBSMCT) Algorithm:

The flow chart for the aforementioned PBSMC approach is depicted in Fig.5.

1) For the current MB if SSE(S1) is less than threshold SSE.
   a. Consider only the inner 5 points of the central diamond for computation.
   b. Obtain the best match SSE(S2) from this five points.
   c. If S2 < S1, the best match is S2 transmit vector location of S2 and go to step 4**.**
   d. If S2 > S1 go tostep3.
2) If SSE (S1) is greater than threshold SSE, consider all 21 search points for the pattern shown in Fig.2. For the first iteration and calculate the SSE at every point proceeding sequentially from the central diamond point to outer points.

Case i) If any of these points has SSE value less than or equal to the threshold SSE transmit vector location of that point and go to step 4**.**

Case ii) If match is found in the central diamond region, continue the search by using EPS and find the best match (threshold condition applies) and transmit vector location of that point and go to step 4**.**

Case iii) If match is found in the outer 8 points continue the search by using the DS and find the best match (threshold condition applies) and transmit vector location of that point and go to step 4**.**

3) Transmit the mean value calculated for the current MB.
4) End.

V. Simulation And Experimental Result

The PBSMC algorithm is simulated for both the cases that are with and without threshold using the luminance component of twenty four CIF/QCIF sequence with variable frame numbers

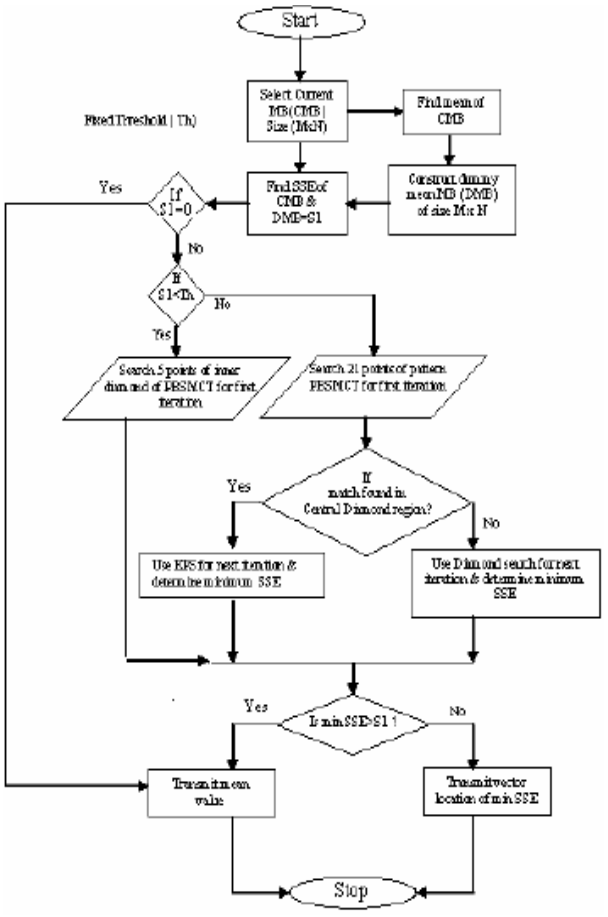

Fig.5. Flowchart of PBSMC (PBSMCT) Algorithm

Out of twenty four sequences, we have listed five video sequences which present different kinds of motion to test the algorithm behavior under different conditions.

1) Miss America: Typical videoconference sequence Fig.8 with head and shoulder movement with fixed background.
2) Akiyo: Videoconference sequence Fig.9 with movement restricted to the face area of the speaker and varying background.
3) Foreman: Sequence Fig.10 with large motion in all direction.
4) Garden: Sequence Fig.11 consists mainly of stationary objects, but with a fast camera panning motion.
5) Coastguard: Sequence Fig.12 with large but uniform motion, almost in the horizontal direction.

The average peak signal to noise ratio PSNR of all encoded pictures is used as a measure of objective quality. The sum of square error (SSE) distortion function is used as the block distortion measure (BDM). We compared the PBSMC against four other block based motion estimation

Methods—FS, TSS, FSS and DS using the following criteria

1) Failure of sub-optimal algorithm to find true motion vector at particular point which FS has found.
2) Average peak signal to noise ratio (PSNR).
3) Average number of search points per block

TABLE. I. PERCENTAGE MOTION VECTOR PROBABILITY DISTRIBUTION USING CIF/QCIF SEQUENCE FOR MULTIPLE SEQUENCES

Hor / Ver	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-7	0.07	0.03	0.03	0.04	0.03	0.03	0.04	0.09	0.03	0.03	0.03	0.03	0.02	0.02	0.06
-6	0.03	0.02	0.02	0.02	0.01	0.02	0.02	0.05	0.02	0.02	0.02	0.02	0.02	0.01	0.03
-5	0.03	0.02	0.02	0.02	0.01	0.02	0.03	0.06	0.03	0.02	0.02	0.02	0.02	0.01	0.03
-4	0.04	0.02	0.02	0.03	0.02	0.03	0.03	0.09	0.04	0.03	0.02	0.02	0.02	0.02	0.04
-3	0.37	0.03	0.02	0.02	0.03	0.03	0.04	0.12	0.05	0.04	0.03	0.02	0.03	0.03	0.07
-2	2.22	0.35	0.03	0.03	0.04	0.06	0.09	0.24	0.09	0.08	0.07	0.06	0.06	0.06	0.11
-1	0.53	1.37	0.58	0.60	0.40	0.16	0.70	2.28	0.71	0.24	0.14	0.11	0.10	0.09	0.21
0	0.05	1.61	2.03	2.34	1.42	1.00	5.79	54.57	3.82	1.11	0.57	0.43	0.37	0.28	0.66
1	0.03	0.61	0.67	0.59	0.30	0.19	0.69	2.66	0.79	0.21	0.13	0.12	0.10	0.08	0.16
2	0.05	0.04	0.04	0.05	0.05	0.05	0.08	0.25	0.21	0.10	0.05	0.06	0.06	0.04	0.08
3	0.03	0.02	0.03	0.03	0.03	0.03	0.03	0.13	0.07	0.06	0.04	0.03	0.05	0.03	0.05
4	0.03	0.02	0.02	0.02	0.02	0.02	0.03	0.08	0.04	0.05	0.04	0.05	0.05	0.05	0.06
5	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.06	0.03	0.05	0.04	0.02	0.04	0.06	0.07
6	0.03	0.02	0.02	0.02	0.02	0.02	0.02	0.06	0.02	0.02	0.02	0.01	0.02	0.03	0.04
7	0.05	0.03	0.03	0.03	0.03	0.02	0.02	0.08	0.03	0.02	0.02	0.02	0.02	0.02	0.06

TABLE. II PERCENTAGE OF TIMES THREE-STEP SEARCH HAVE FAILED TO FIND BEST MATCH THAT FULL SEARCH HAS FOUND AT PARTICULAR POINT WITHIN 15 X 15 SEARCH WINDOW FOR 75 FRAMES OF AKIYO SEQUENCE.

	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-7	70.2	74.72	80.91	44.73	74.63	72.66	76.94	56.13	68.62	72.8	70.62	69.77	74.1	74.12	73.96
-6	81.94	65.27	75.2	53.06	71.19	56.02	80.48	45.7	70.45	58.33	78.64	62.36	75	65.18	74.94
-5	74.7	77.37	48.96	44.41	59.17	64.06	71.49	32.45	60.84	69.04	63.19	48.49	59.63	67.09	72.13
-4	76.91	59.85	66.58	**0**	54.4	53.9	62.34	**0**	56.59	51.45	57.7	**0**	56.68	57.87	76.2
-3	69.49	69.04	67.52	56.8	36.65	58.43	65.11	24.5	54.72	53.05	56.37	52.76	57.91	65.78	68.84
-2	77.07	58.7	69.9	46.58	60.4	13.92	67.01	16.75	54.35	24.41	55.66	54.45	59.17	50.39	69.39
-1	73.84	59.82	34.14	22.95	38.29	69.19	19.72	11.74	20.43	54.98	50.27	51.35	53.83	66.38	76.71
0	60.25	53.44	15.29	**0**	16.12	14.64	8.72	**0**	11.14	21.3	23.74	**0**	23.23	37.82	46.55
1	76.54	67.85	45.31	22.85	35.05	61.5	17.56	12.95	22.85	67.57	65.14	60.22	59.14	70.18	79.36
2	78.83	66.33	75.61	56.18	67.25	17.01	58.4	18.44	46.19	22.29	56.5	51.1	56.71	56.11	70.17
3	74.15	76.1	65.33	50.85	56.8	53.52	57.56	24.78	47.87	47.54	37.5	48.18	40.79	54.3	65.09
4	77.15	65.58	61.95	**0**	60.96	51.88	56.42	**0**	57.21	47.89	42.01	**0**	29.37	41.74	65.26
5	75.83	74.63	62.8	44.65	61.09	67.12	64.2	28.84	65.5	54.98	42.55	49.58	29.41	55.58	61.69
6	80.4	68.38	80.78	57.78	78.06	65.65	78.73	51.73	78.7	52.96	63.94	56.64	67.17	46.5	68.17
7	73.76	75.32	88.25	60.78	79.75	70.66	75	55.73	78.19	73.96	79.81	68.03	75.18	78.33	65.23

TABLE. III PERCENTAGE OF TIMES FOUR-STEP SEARCH HAVE FAILED TO FIND BEST MATCH THAT FULL SEARCH HAS FOUND AT PARTICULAR POINT WITHIN 15 X 15 SEARCH WINDOW FOR 75 FRAMES OF AKIYO SEQUENCE

	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-7	84.35	86.91	84.69	86.01	81.59	76.35	78.25	65.87	74.5	76.13	80.57	83.63	86.41	87.33	85.67
-6	90.74	83.68	96.4	80.82	95.34	71.78	95.72	64.9	97.1	72.84	96.59	83.65	97.18	89.73	87.34
-5	86.02	98.17	85.67	88.18	83.49	79.69	88.26	76.92	85.62	83.29	87.83	89.63	91.48	97.44	84.07
-4	85.45	82.2	92.09	53.5	77.36	54.16	78.34	37.44	69.43	48.94	77.02	61.48	89.32	79.53	87.54
-3	78.84	94.31	85.19	75.2	39.96	51.61	59.21	25.77	53.41	40.68	57.88	75.59	82.24	97.34	77.6
-2	77.96	67.63	86.87	46.8	55.78	**0**	60.6	**0**	42.24	**0**	39.42	36.97	78.63	61.79	76.07
-1	78.52	93.32	75.59	52.97	42.25	63.03	14.6	9.05	12.88	36.63	43.92	65.32	82.16	96.75	81.44
0	66.43	64.83	67.13	38.49	19.01	**0**	6.51	**0**	8.01	**0**	24.87	35.69	67.85	48.98	52.72
1	82.41	96.61	82.08	61.3	44.13	52.09	13.7	11.21	17.46	56.31	60.72	71.06	82.99	96.39	80.32
2	84.19	77.72	87.95	59.24	63.22	**0**	45.05	**0**	30.9	**0**	46.13	27.48	73.5	50.57	58.21
3	82.64	94.78	87.19	74.94	65.31	49.47	55.04	24.68	38.86	32.57	41.83	65.64	73.55	95.43	75.26
4	87.33	89.02	94.39	64.86	79.41	53.62	72.04	36.29	74.88	35.36	65.28	45.95	79.85	62.67	77.07
5	88.03	95.22	91.67	88.68	89.08	84.75	86.11	73.68	84.99	69.05	60.47	82.82	76.83	95.69	72.5
6	90.45	89.34	98.7	83.23	98.39	77.1	96.64	67.89	93.77	64.51	97.4	82.74	94.34	70.56	81.98
7	86.94	87.98	93.61	87.17	85.93	80.84	80.1	65.6	82.63	78.39	85.9	84.01	84.67	89.17	79.98

TABLE. IV PERCENTAGE OF TIMES DIAMOND SEARCH HAVE FAILED TO FIND BEST MATCH THAT FULL SEARCH HAS FOUND AT PARTICULAR POINT WITHIN 15 X 15 SEARCH WINDOW FOR 75 FRAMES OF AKIYO SEQUENCE.

	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-7	81.91	81.72	80.53	74.09	77.36	77.09	76.44	66.96	73.21	78.86	81.52	83.38	84.87	87.06	80.2
-6	89.16	86.19	84.8	81.22	80.93	82.57	83.96	68.2	81.27	77.47	86.69	87.83	91.94	90.63	85.86
-5	86.27	91.13	77.91	80.83	75.69	74.06	76.31	64.5	69.47	75.89	83.48	82.94	87.41	88.89	84.54
-4	81.76	84.47	87.24	65.42	78.93	66.5	65.16	43.35	60.81	63.39	75.98	75.68	82.79	87.4	84.19
-3	76.61	81.85	76.64	77.33	47.88	62.05	43.96	26.1	41.86	50.96	59.83	79.53	79.08	79.28	79.36
-2	76.04	73.67	69.9	56.95	58.82	24.27	38.31	**0**	23.56	20.83	44.86	50.74	63.21	66.6	77.35
-1	75.26	69.81	61.56	51.43	32.88	36.53	**0**	7.14	**0**	23.12	32.02	53.58	58.27	65.37	65.11
0	69.31	67.57	53.83	42.42	18.11	**0**	4.17	**0**	5.79	**0**	24.44	38.94	46.05	51.59	54.67
1	70.53	70.19	62.81	50.16	35.41	35.66	**0**	6.7	**0**	26.37	34.23	49.81	54.52	64.95	64.82
2	83.71	82.75	80.92	66.75	64.36	33.41	29.75	**0**	16.6	14.8	48.87	37.2	53.76	58.24	61.92
3	82.08	90.93	83.17	77.13	64.91	59.91	43.02	24.17	25.43	40.73	45.67	63.15	57.6	72.03	72.25
4	87.1	91.69	88.29	80.83	77.54	64.35	65.74	41.93	56.89	44.46	62.5	54.05	65.78	63.02	72.15
5	86.03	90.81	87.2	87.42	84.64	76.95	70.99	56	72.71	51.46	54.1	72.39	63.46	72.15	70.04
6	88.19	89.34	91.21	88.02	83.23	85.11	82.09	70.9	81.56	71.55	78.44	86.28	85.28	71.73	77.18
7	80.59	87.34	89.9	85.69	76.54	79.64	77.81	65.91	79.73	79.43	85.9	82.53	85.4	88.33	79.35

Fig. 6 Miss America Sequence

Fig.7 Akiyo sequence

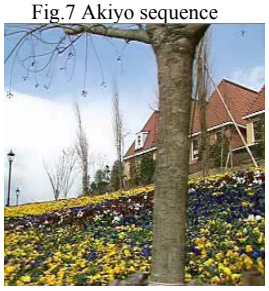

Fig.8 Foreman sequence

Fig. 9Garden sequence

Fig.10 Coastguard sequence

We know that TSS, FSS, and DS algorithm have inherent problem in their search pattern due to which these algorithms have poor PSNR as compared to FS. In order to analyze the shortcomings in the search pattern of these algorithms we have considered three cases i) MV located in central diamond region. ii) MV located in proximity of central diamond region. iii) MV located at the edge of search window. The carphone sequence is considered for the first case, the SSE values of which are derived using FS for frame 45 for MB at (15,1) and are depicted in Table V. As FS method computes SSE at all points in search window of 15x15, it is very easy to see that minimum SSE is at location (-1,0), but TSS is not able to find the minimum value instead gets trapped in to local minima at location (4,4) and finds minimum SSE point at (5,5), which result in large residual error and poor PSNR, FSS similar to TSS and DS similar to TSS gets trapped in to local minima at (2,2) and (1,1) and finds minimum SSE point at locations(5,5), (0,1) respectively resulting in poor PSNR. In other words due to its static search pattern TSS, FSS and DS fails to locate the minimum SSE point located at (-1,0) in central diamond region, but the above limitation of getting trapped in local minima is overcome by PBSMC algorithm which finds the minimum SSE point at location (-1,0) in very first iteration.

The search patterns of TSS, FSS, DS and PBSMC to locate minimum SSE point for case(i) is as tabulated in Table VI. The first iteration for TSS and FSS as shown in first column is comparison, in terms of SSE at nine different locations with step size of ± 4 pixels and ± 2 pixel respectively from the center point. The minimum cost function (SSE) out of this nine point becomes center point for the next iteration, in this case locations (4,4), (2,2) are minimum SSE points for TSS and FSS in first iteration. The process to find minimum SSE

will continues till the step size is reduced to ± 1 pixel. The search sequence in case of DS is also nine different points forming diamond shape with search location as shown in first column of Table VI, the further steps of DS algorithm to locate minimum SSE points are almost similar to FSS. As can be seen TSS and DS converges to minimum SSE value in third iteration with 25 maximum search points and FSS converges to minimum SSE value in fourth iteration with maximum of 33 search points, the important point to note here is that these three algorithms uses considerable search points, still are unable to find true MV located at location (-1,0). Our algorithm PBSMC due to its adaptive search pattern takes only few points to track true motion vector that FS has searched.

As we have seen the limitation of case(i) now let us analyze case(ii), the sequence under consideration is stefan, which is large motion sequence involving camera panning and zooming movements, the MB for comparison of minimum cost function is located at (29,8) for frame number 45. SSE value for all the points within search window of 15x15 using full search method is computed and is as shown in Table VII, the minimum SSE is found at location (1,-4) which is between central point (0,0) and edge of the search window. Here FSS and DS gets trapped at (0,-2) and finds minimum SSE point of value at location (0,-3), whereas TSS miserably gets misguided and travels in opposite direction to locate minimum SSE point at location (-7,3) which result in poor PSNR, but PBSMC algorithm in fourth iteration finds minimum SSE point at same point which FS has found.

As mentioned earlier that the sub optimal algorithms are not efficient to find true MV, in other words if TSS can track large motion it is unable to track low motion whereas FSS and DS can track low motion but are unable to track large motion although DS performs better than FSS in the vicinity of central point but fails to track MV located at edge of the window. The following example which considers third case with frame number 128 of bus sequence, the MB placed at (9,21) has minimum SSE at (5,-5) which is found by using FS method. As shown in Table VIII, the FSS and DS totally gets misguided and are unable to track true motion vector which is located at (5,-5) instead FSS finds minimum SSE value at location (-2,7) and DS finds minimum SSE value at location (-2,6) both of which are in opposite direction as compared to minimum SSE vector location (5,-5), TSS comes closer (1,-5) to vector location that FS has found (5,-5) but still is unable to detect the true motion vector. It is important to note that our algorithm tracks the true motion vector in fourth iteration due to search points located in outer diamond region which justify our selection of additional search points in outer diamond.

The Search sequence of TSS, FSS, DS and PBSMC in Table VIII is tabulated in Table IX, along with their minimum SSE point for third cases that is minimum SSE point located at the edge of search window. As can be seen from Table IX that TSS converges to minimum SSE value in third iteration with 25 maximum search points at location (5,5) and FSS and DS converges to minimum SSE value in fourth iteration with maximum of 33 search points. Again these three algorithms fail to find true MV located at location (-1,0). Our algorithm PBSMC due to its intelligent search pattern takes only few points to track true motion vector that FS has searched. It can be concluded from above observation that TSS algorithm is unable to detect motion vector in proximity of central diamond point and FSS and DS are unable to detect motion vector located at the edge of search window but with PBSMC we are able to track almost all MVs that is MVs located in the central diamond region, in proximity of central diamond point or at the edge of the search window.

Tables X and X1 summarizes the PSNR value and average number of search points for algorithms like FS, probability based search with mean correction (PBSMC), probability based search with mean correction and threshold (PBSMCT), DS, FSS, and TSS. Simulation result shows considerable improvement in PSNR of sequences like suzie, carphone, mother and daughter, silent, tempete and waterfall.

TABLE. V FAILURE OF TSS, FSS, AND DS SEARCH PATTERNS TO LOCATE MINIMUM SSE POINT IN CENTRAL DIAMOND REGION FOR CARPHONE SEQUENCE, FRAME 4, ROW-15, COL-11

	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-7	4255	919	4645	6697	7588	7920	13555	31224	59399	93597	133642	178359	228128	280061	330412
-6	4455	4406	3409	4757	5692	6401	11593	28582	57652	98205	147048	202957	265198	328758	389977
-5	3944	4351	4871	4934	5082	6030	12265	28798	57327	99158	153395	218320	289798	363243	433032
-4	4072	3763	3896	4300	4429	4216	10609	26711	53245	93849	148910	217923	295717	381436	463323
-3	4334	3804	3627	3521	3432	723	2665	17550	45068	85135	139056	209140	287847	377169	465302
-2	4569	4169	3828	3473	3656	3162	201	5014	23701	55345	99324	159628	230121	313606	405815
-1	5013	4442	4074	3724	3408	3357	2542	99	7188	29486	64411	114008	175004	249796	337246
0	5413	4625	4438	3975	3364	3065	3479	2204	282	10191	35561	74917	125502	189687	268627
1	5293	4774	4640	4347	3670	3376	3220	3390	1472	967	13388	41042	79795	131766	197125
2	5384	4840	4639	4421	3815	3471	3271	3081	3107	1052	2014	16094	42625	82669	136297
3	5406	5061	4889	4289	3893	3719	3345	3249	3122	3163	826	3013	16754	44445	86988
4	5296	5134	4932	4507	4102	3821	3705	3404	3367	3374	3241	593	3843	18914	49413
5	5274	4890	4690	4474	4175	3741	3538	3503	3469	3432	3528	2709	541	5388	24042
6	5103	4784	4507	4317	4169	3781	3659	3723	3739	3857	3675	3521	2468	569	9083
7	5045	4859	4539	4517	4275	3841	3933	3827	4053	3892	3908	3725	3626	1788	1534

TABLE. VI SEARCH ITERATION AND MINIMUM SSE VALUE FOR TSS, FSS, DS AND PBMCT ALGORITHM FOR CARPHONE SEQUENCE, FRAME-4 ROW-15, COL-11

Sr.no	Algorithm	Search points in different iterations with minimum SSE value			
		1st iteration	2nd iteration	3rd iteration	4th iteration
1	TSS	(0,0)(-4,0)(-4,-4) (-4,4)(0,-4)(4,-4) (4,0)(4,4)(0,4) [SSE-593 (4,4)]	(2,2)(2,4)(2,6) (4,2)(4,0)(4,6) (6,2)(6,4)(6,6) [SSE-569 (6,6)]	(5,5)(5,6)(5,7) (6,5)(6,6)(6,7) (7,5)(7,6)(7,7) [SSE-541(5,5)]	
2	FSS	(0,0)(-2,0)(-2,-2) (-2,2)(0,-2)(0,2) (2,-2)(2,0)(2,2) [SSE-1052(2,2)]	(0,0)(0,2)(0,4) (2,0)(2,2)(2,4) (4,0)(4,2)(4,4) [SSE-593(4,4)]	(2,2)(2,4)(2,6) (4,2)(4,4)(4,6) (6,2)(6,4)(6,6) [SSE-569(6,6)]	(5,5)(5,6)(5,7) (6,5)(6,6)(6,7) (7,5)(7,6)(7,7) [SSE-541(5,5)]
3	DS	(0,0)(-2,0)(-1,-1) (-1,1)(0,-2)(0,2) (1,-1)(1,1)(2,0) [SSE-1472(1,1)] (Local minima)	(-1,1)(0,-1)(0,1) (1,-1)(1,1)(1,3) (2,0)(2,2)(3,1) [SSE-1472(1,1)]	(0,0)(0,1)(0,2) (1,0)(1,1)(1,2) (2,0)(2,1)(2,2) [SSE-282(0,1)]	
4	PBSMC (PBSMCT)	(0,0)(0,-1)(0,-2)(0,1) (0,2)(-1,0)(-1,-1)(-1,1) (-2,0)(1,0)(1,-1)(1,1) (2,0)(-6,0)(6,0)(-3,-3) (-3,3)(3,-3)(3,3)(0,-6) (0,6) [SSE-99(-1,0)]	IF S1=0 search iteration are not needed. IF S1< TH search inner 5 points (0,0)(0,-1)(0,1) (-1,0)(1,0). IF S1> TH search all 21 points		
The minimum SSE for above MB located at (15,11) using FS is 99. PBSMC algorithm is able to detect true value located in central diamond region and all other algorithm fails to locate the true motion vector.					

TABLE. VII FAILURE OF TSS, FSS, AND DS SEARCH PATTERNS TO LOCATE MINIMUM SSE POINT IN PROXIMITY OF ITS INITIAL SEARCH REGION FOR STEFAN SEQUENCE, FRAME 45, ROW-28, COL-8

	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-7	25583	25879	25392	24973	25024	24892	26150	28272	26771	19490	12575	13215	23481	32110	36829
-6	25764	25914	25668	25910	25789	26153	26559	23637	17593	13637	19181	29162	32172	30642	42376
-5	25050	25257	25204	25454	25536	25892	22824	15836	13215	22366	38628	46853	41298	35311	46953
-4	24395	24420	24088	23533	24121	24906	22769	19798	20584	28877	39174	45274	50464	54866	58714
-3	24698	24515	23914	22966	24309	26658	26275	25077	24603	26758	35369	48064	63793	77234	83100
-2	24533	24311	25613	28335	29221	27133	27687	34756	38349	38038	43674	53929	65304	80819	100205
-1	23986	25304	29864	31416	23756	14702	13138	30895	53746	68263	73814	75614	78370	93906	120299
0	25209	27866	28641	20018	11239	13044	19071	32219	56520	87845	108711	117530	120182	132104	144522
1	26796	25401	17377	10479	20078	44048	65925	83090	97309	117584	136440	152192	167763	176026	167742
2	24942	20022	15723	24046	50501	83356	112218	140275	154293	160607	160489	162161	169776	175800	165370
3	22349	19854	28226	46624	69385	93127	118947	152717	177897	189814	180117	163170	153913	156655	151166
4	24001	26914	42598	55870	64955	80609	103047	134671	164004	182292	175382	162913	161541	164673	154691
5	31539	35627	48284	59347	72397	92895	113031	133676	153708	168502	162936	158592	165170	165585	149119
6	36276	36785	44159	64326	95361	125007	147484	167556	183691	192413	183517	173912	169304	158416	135781
7	31911	26693	31490	53072	83013	109511	134787	157981	172922	179097	168068	150823	137565	124837	103309

TABLE. VIII FAILURE OF TSS, FSS, AND DS SEARCH PATTERNS TO LOCATE MINIMUM SSE POINT CLOSER TO EDGES OF SEARCH WINDOW FOR BUS SEQUENCE, FRAME 128, COL-9, COL-21.

	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
-7	105324	57114	7014	15421	98009	192661	226532	223813	226121	199138	143396	106323	99960	94692	91818
-6	106727	58023	7174	14953	97239	193232	228183	224768	225351	197830	142790	106299	100286	96003	94398
-5	107535	58227	7012	14769	96541	193815	230307	226461	225349	196707	141783	104495	97761	94587	94069
-4	107519	57532	6566	15003	97226	194977	231202	226140	224418	195835	140566	102195	95317	93394	93127
-3	107392	57283	6613	15185	97789	195684	231135	225271	223306	195221	139800	100826	93586	91651	91202
-2	107986	57379	6677	15246	96866	194496	231635	226881	223545	194131	139238	100793	93406	91208	89338
-1	108451	56879	6260	15117	95802	193229	232307	228483	223255	192142	138764	102964	97676	95620	90581
0	107314	55970	5933	14892	95136	193387	233506	**229977**	223601	192053	139979	106219	102027	98480	90686
1	106250	55421	5890	15253	95518	194126	235397	232503	225740	193928	141724	107437	100753	95039	88083
2	106018	55100	5744	16149	96380	194721	236503	233640	226926	194494	141433	106235	98273	91945	86182
3	105209	54662	5702	16775	96955	195581	237306	233890	226986	193798	141265	107479	101030	94736	88710
4	103441	53621	5412	16926	95846	193868	236827	234817	227605	193226	142378	110615	104081	96607	90747
5	101952	52471	**5271**	17460	94300	190990	236306	236943	230375	195032	144870	112828	105316	97381	91727
6	99315	50974	5598	18668	93886	189370	235775	237693	232154	197029	147101	113383	104498	96748	92879
7	92881	48162	7137	22128	94389	184221	227503	229034	223906	190882	146692	115247	104428	96422	94897

The PSNR value of this sequences are even better than full search with less number of search points, this is because of mean correction technique used to improve the quality of the picture. Other sequences like akiyo, coastguard, container, etc PSNR is comparable to FS but far better than TSS, FSS, and DS. Algorithm like PBSMC uses almost same number of search points that TSS uses but PSNR values are better than TSS, FSS and DS algorithm.

TABLE. IX SEARCH ITERATION AND MINIMUM SSE VALUE FOR TSS, FSS, DS AND PBSMCT ALGORITHM FOR BUS SEQUENCE, FRAME-128 ROW-9, COL-21

Sr.no	Algorithm	Search points in different iterations with minimum SSE value			
		1st iteration	2nd iteration	3rd iteration	4th iteration
1	TSS	(0,0)(-4,0)(-4,-4) (-4,4)(0,-4)(4,-4) (4,0)(4,4)(0,4) [SSE-14892 (0,-4)]	(-2,-6)(-2,-4)(-2,-2) (0,-6)(0,-4)(0,-2) (2,-6)(2,-4)(2,-2) [SSE-14892 (0,-4)]	(-1,-5)(-1,-4)(-1,-3) (0,-5)(0,-4)(0,-3) (1,-5)(1,-4)(1,-3) [SSE-5890(1,-5)]	
2	FSS	(-2,-2)(-2,0)(-2,0) (0,-2)(0,0)(0,2) (2,-2)(2,0)(2,2) [SSE-192053(0,2)]	(-2,0)(-2,2)(-2,4) (0,0)(0,2)(0,4) (2,0)(2,2)(2,4) [SSE-100793(-2,4)]	(-4,2)(-4,4)(-4,6) (-2,2)(-2,4)(-2,6) (0,2)(0,4)(0,6) [SSE-91208(-2,6)]	(-3,5)(-3,6) (-3,7)(-2,5) (-2,6)(-2,7) (-1,5)(-1,6) (-1,7)[SSE-89338(-2,7)]
3	DS	(0,0)(-2,0)(-1,-1) (-1,1)(0,-2)(0,2) (1,-1)(1,1)(2,0) [SSE-192053(0,2)]	(0,2)(-1,1)(-1,3) (0,0)(0,4)(1,1) (1,1)(2,2)(-2,2) [SSE-106219(0,4)]	(0,4)(-1,3)(-1,5) (0,2)(0,6)(1,3) (1,5)(2,4)(-2,4) [SSE-97676(-1,5)]	(-2,4)(-2,5) (-2,6)(-1,4) (-1,5)(-1,6) (0,4)(0,5) (0,6) [SSE-91208]
4	PBSMC (PBSMCT)	(0,0)(0,-1)(0,-2)(0,1) (0,2)(-1,0)(-1,-1)(-1,1) (-2,0)(1,0)(1,-1)(1,1) (2,0)(-6,0)(6,0)(-3,-3) (-3,3)(3,-3)(3,3)(0,-6) (0,6) (-6,0)(-3,-3) (-3.3)(0,-6)(3,-3)(3,3) (6,0) [SSE-55970(0,-6)]	(0,-6)(-2,-6)(-1,-7) (-1,-5)(1,-7)(1,-5) (2,-6)(0,-4) [SSE-5890(1,-5)]	(1,-5)(-1,-5) (0,-6)(0,-4) (1,-7)(1,-3) (2,-6)(2,-4) (3,-5) [SSE-5702(3,-5)	(3,-5)(1,-5) (2,-6)(2,-4) (3,-7)(3,-3) (4,-6)(4,-4) (5,-5)[SSE-5271(5,-5)]
The minimum SSE for above MB located at (5,-5) using FS is 5271. PBSMC algorithm is able to detect true value located at edge of search window and all other algorithm fails to locate the true motion vector.					

TABLE. X COMPARISON OF THE AVERAGE PSNR (DB) VALUES OF FS, PBSMC, PBSMCT, DS, FSS AND TSS

Sequence	FS	PBSMC	PBSMCT	DS	FSS	TSS
Miss America	41.866	41.878	41.84	41.78	41.618	41.587
Suzie	37.422	37.431	37.428	37.157	37.006	36.89
Akiyo	44.776	44.772	44.772	44.763	44.665	44.647
Carphone	33.654	33.672	33.668	33.106	32.971	32.99
Coastguard	33.804	33.8	33.8	33.688	33.59	33.173
Container	43.271	43.274	43.269	43.187	43.202	43.226
Foreman	33.919	33.751	33.748	33.068	32.998	32.989
Hall Monitor	37.698	37.695	37.693	37.666	37.622	37.589
Mobile	26.332	26.328	26.328	26.302	26.304	26.296
Mthr &	42.185	42.217	42.201	42.083	42.022	41.807
News	37.923	37.878	37.877	37.652	37.571	37.451
Silent	37.555	37.594	37.594	37.17	37.094	37.028
Salesman	41.248	41.244	41.241	41.192	41.13	41.064
Claire	43.362	43.355	43.34	43.342	43.307	43.27
Grandma	43.673	43.677	43.647	43.645	43.632	43.632
Highway	36.376	35.891	35.89	35.579	35.398	35.385
Bridge (Close)	38.576	38.576	38.575	38.575	38.575	38.574
Bridge (Far)	41.648	41.641	41.64	41.616	41.611	41.611
Stefan	27.32	26.647	26.647	25.055	25.128	25.907
Bus	26.808	25.316	25.316	22.85	22.758	24.062
Flower	27.204	27.116	27.115	26.909	26.53	24.038
Tempete	27.926	28.213	28.213	27.709	27.661	27.694
Waterfall	34.794	34.794	34.794	34.792	34.791	34.767
Paris	35.174	35.119	35.119	34.79	34.686	34.625

TABLE. XI COMPARISON OF AVERAGE NUMBER OF SEARCH POINTS OF FS, PBSMC, PBSMCT, DS, FSS AND TSS

Sequence	FS	PBSMC	PBSMCT	DS	FSS	TSS
Miss America	204.28	22.47	6.88	13.69	16.756	23.268
Suzie	204.28	22.11	14.51	13.49	16.579	23.256
Akiyo	204.28	10.56	4.31	12.22	15.816	23.212
Carphone	204.28	23.81	17.31	14.71	17.131	23.322
Coastguard	204.28	22.48	20.44	14.16	17.03	23.286
Container	204.28	20.76	4.83	12.33	15.883	23.218
Foreman	204.28	24.79	20.32	15.7	17.71	23.316
Hall Monitor	204.28	21.11	16.85	12.65	16.118	23.254
Mobile	204.28	20.96	20.55	12.33	15.869	23.216
Mthr &	204.28	21.78	8.08	13.15	16.493	23.261
News	204.28	15.38	6.01	12.57	16.017	23.212
Silent	204.28	21.52	8.6	12.94	16.227	23.218
Salesman	204.28	20.73	6.4	12.29	15.858	23.214
Claire	204.28	19.58	4.78	12.42	15.926	23.231
Grandma	204.28	21.42	5.87	12.84	16.276	23.266
Highway	204.28	21.7	20.43	13.72	16.801	23.336
Bridge (Close)	204.28	18.59	17.57	12.22	15.829	23.225
Bridge (Far)	204.28	21.16	19.91	12.65	16.048	23.218
Stefan	214.52	27.46	22.34	17.11	18.733	24.141
Bus	214.52	31.75	31.5	20.08	20.388	24.312
Flower	214.52	25.7	18.07	16.7	19.172	24.157
Tempete	214.52	23.55	20.28	14.27	17.202	24.141
Waterfall	214.52	21.58	20.49	12.7	16.431	24.099
Paris	214.52	21.71	8.92	12.95	16.605	24.099

VI. CONCLUSION

Our proposed algorithm, PBSMC, has been presented in this paper for motion estimation. The proposed algorithm uses stochastic approach to locate the true motion vector. Simulation results show that PBSMC achieves better estimate accuracy as compared to earlier proposed algorithms. Due to its efficient search pattern it track motion vector in vicinity of central point as well as at edge of the search window which makes it more applicable search algorithm for video with small and large motion. Mean correction technique developed, considerably improves PSNR at much reduced computational complexity. The early termination of algorithm with reasonable PSNR is possible using PBSMCT algorithm, using the threshold and mean correction technique. The computations involved in this technique are considerably

low as compared to FS, TSS, FSS, DS, and PBSMC with PSNR comparable to PBSMC and much better than other sub optimal algorithm. In addition proposed algorithm is more robust as compared to earlier algorithms because it is flexible enough to work well, for any search range and window size which will be useful in rate constrained environment. Even the performance of PBSMC is consistent for the image sequence that contains complex movements such as camera panning and zooming. The simulation result demonstrates that the proposed algorithm is very suitable for high quality video encoding.

REFERENCES

[1] B. Haskell, A. Puri, A.Netravali, "Digital video: An introduction to MPEG-2", Kluwer Academic Publisher, 2000.
[2] K.Jack, "Video demystified: A Handbook for the Digital Engineer", Elsevier Publisher, Linacre House, Oxford, 2005
[3] T.Koga,K.Iinuma,A.Hirano,Y.Iijima, and T.Ishiguro,"Motion compensated interframe coding for video conferencing," in Proc. National Telecommunications Conf., Nov. 29–Dec. 3, 1981, pp. G.5.3.1–G.5.3.5.
[4] Li and E Salari, "Successive Elimination Algorithm for Motion Estimation", IEEE Transactions on Image Processing, vol. 4, no. 1, pp. 105-107, Jan.1995.
[5] B.Zeng, and M.Liou, "A New Three Step Search Algorithm for Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, no.4, pp. 438–442, Aug. 1994.
[6] Lai-Man Po and Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 313317, June 1996.
[7] Kuan-Tsang wang , "Efficient Four-Step Search'', IEEE international symposium on Circuits and Systems, 1998, ISCAS apos,'98 vol. 4, pp. 217-220, June 1998.
[8] Jo Yew Tham, Surendra Ranganath, Maitreya Ranganath and Ashraf Ali Kassim, "A Novel Unrestricted Centre Biased Diamond Search Algorithm for Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 4, pp. 369-377, Aug. 1998.
[9] M.Ghanbari, "The cross search algorithm for motion estimation," IEEE Trans.commun.,vol.38, no.7, pp.950-953, July 1990.
[10] Jong-Nam Kim, Sung-Cheal Byun, Yong-Hoon Kim, and Byung-Ha Ahn, "Fast Full Search Motion Estimation Algorithm using Early Detection of Impossible Candidate Vectors", IEEE Transaction on Signal Processing, vol. 50, no. 9, pp.2355-2365, Sept. 2002.
[11] Antonio Chimienti, Claudia Ferraris, and Danilo Pau, "A complexity-bounded motion estimation algorithm", IEEE Transactions on image processing, vol. 11, no. 4, pp.387-392, Apr. 2002.
[12] B.Liu and A. Zaccartin, "New Fast Algorithms for Estimation of Block Motion Vectors," IEEE Trans. Circuits Syst. Video Technol., vol. 3, no.2, pp.148– 157, Apr.1993.
[13] L.W.Lee, J.F. Wang, J.Y.Lee, and J.D. Shie, " Dynamic Search Window Adjustment and Interlaced Search for Block-Matching Algorithm, " IEEE Trans. Circuits Syst. Video Technol., vol.3, no.1, pp.85-87, Feb.1999
[14] R. Srinivasan and K.R. Rao, "Predictive Coding Based on Efficient Motion Estimation," IEEE Trans. Circuits Syst.Video Technol., vol.Comm-33, no.8, pp.888-896, Aug.1985.
[15] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol. COM-29, no.12, pp.1799–1808, Dec.1981.
[16] Hanan, suneer, mohsen and magdy bayoumi, "A new efficient block-matching algorithm for motion estimation" Journal of VLSI Signal processing system, vol. 42, issue.1, pp.21-33, Jan. 2006.
[17] Z. Zhou, M.T. Sun, S. Hsu, "Fast Variable Block-size Motion Estimation Algorithms Based on Merge and Split Procedure for H.264/MPEG-4 AVC," IEEE ISCAS Conference. 2004.
[18] Ishfaq Ahmad, Weiguo Zheng, Jiancong Luo and Ming Liou "A Fast Adaptive Motion Estimation Algorithm" IEEE Trans. Circuits Syst. Video Technol., vol.16, no.3, pp.420-438, Mar.2006.
[19] A. M. Tourapis, O. C. Au, and M. L. Liou, "Fast block-matching motion estimation using advanced predictive diamond zonal search (APDZS), ISO/IEC JTC1/SC29/WG11 MPEG2000/M5865, Noordwijkerhout, The Netherlands, Mar., 2000.
[20] "Fast block-matching motion estimation using predictive motion vector field adaptive search technique (PMVFAST),"ISO/IECJTC1/SC29/WG11 MPEG2000/M5866, Noordwijkerhout, The Netherlands, Mar. 2000.
[21] J. Feng, K. T. Lo, H. Mehrpour, and A. E. Karbowiak, "Adaptive blockmatching motion estimation algorithm for video coding," IEE Electron. Lett., vol. 31, no.18, pp. 1542–1543, 1995.
[22] B. C. Song and J. B. Ra, "A hierarchical block matching algorithm using partial distortion criteria," in Proc. VCIP Vis. Commun. Image Process.,San Jose, CA, pp.88–95. 1998.
[23] J. B. Xu, L. M. Po, and C. K. Cheung, "A new prediction model search algorithm for fast block motion estimation," in IEEE Int. Conf. ImageProcess., 1997, pp. 610–613. 1997.
[24] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol.13, no.7, pp. 560576, 2003.
[25] Kuo-Liang chung and Lung-chung chang, " A New Predictive Search Area Approach for Fast Block Motion Estimation," IEEE Transaction on Image Processing, vol.12, no.6, pp. 648-652, June 2003.
[26] Ce-Zhu, Wei-Song Qi, and Ser-W, "Predictive Fine Granularity Successive Elimimination for Fast Optimal Block- Matching motion Estimation**,"** IEEE Transaction on Image Processing, vol.14, no.2, pp.213-221 Feb.2005.
[27] Shen-Chen Tai, Ying-Ru Chen, Yu-Hung chen, "Small-diamond-based search algorithm for fast block motion estimation," Elsevier Publisher, Signal Processing: Image Communication, vol. 22, no.10, pp. 877- 890, Nov. 2007

Deepak J.Jayaswal received his B.E degree in electronics engineering from Walchand College of engineering, Shivaji University, Kolhapur, Maharashtra, India in 1991, M-Tech degree in communication from IIT Bombay, 2002. He is student member IEEE. He is working in St. Francis Institute of Technology, Borivali, Mumbai as Assistant Professor in Electronics and Telecommunication department, His research area includes Image and video processing.

Mukesh A. Zaveri received the B.E. degree in electronics engineering from Sardar Vallabhbhai Regional College of Engineering and Technology, Surat, India, in 1990, the M.E. degree in electrical engineering from Maharaja Sayajirao University, Baroda, India, in 1993, and the Ph.D. degree in electrical engineering from the Indian Institute of Technology Bombay, in 2005. He is currently an Assistant Professor and the Head of the Computer Engineering Department, Sardar Vallabhbhai National Institute of Technology. His current research interests include the area of signal and image processing, multimedia, computer networks, sensor networks, and wireless communications.