ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

A New Hybrid Block Based Motion Estimation Algorithm for Video Compression

Piyush N. Korat[1], Dr. Dhaval R. Bhojani[2]

PG student, Dept. of ECE, Darshan Institute of Engineering & Technology, Rajkot, India [1]

Assistant Professor & HOD, Dept. of ECE, Darshan Institute of Engineering & Technology, Rajkot, India [2]

ABSTRACT: The main objective of the motion estimation is to powerfully reduce temporal redundancy between successive frames to achieve significant video compression. Several block-based fast motion estimation algorithms have been proposed in order to improve computational complexity. In this paper different types of block matching algorithms are discussed that range from the very basic to the fast block matching algorithm. This paper suggests a hybrid motion estimation technique (H-MET) based on Diamond Search (DS) and Adaptive Rood Pattern Search (ARPS) algorithms, which effectively detect the slow as well as fast motion with effective computation time and less number of search points compared to DS and ARPS algorithms. The performance of the proposed algorithm is evaluated in terms of matching criteria that is sum of absolute difference (SAD). Experimental results show that the proposed scheme performs better than DS and ARPS by exploiting less number of search points and form compensated frame with better PSNR.

KEYWORDS: Video compression, Motion estimation, Block matching algorithms, SAD, PSNR, DS, ARPS

I. INTRODUCTION

With the beginning of the multimedia age and the wide use of video on demand over internet, video storage and streaming video has become extremely popular. The digital video application has gained wide appeal in mobile terminals. Now a day's video is required in many remote video conferencing systems and for many real time applications (like space application). But a major problem still remains in a digital video, that is size is very large and the memory storage of the devices and the bandwidth of the transmission channel are finite, so it is essential to develop digital video compression methods which can produce better compression ratio as well as preserve quality of reconstructed video in order for the creation of high quality, affordable video products.

Digital video is nothing but a sequence of frames/images. Each frame consists of definite number of pixels. The number of pixels in each frame signifies resolution of the digital video. Resolutions can be vary according to the video format. For example, if we take HDTV video, it is having resolution of 1280 x 720 pixels in each frame, so the data rate would be:

$$\left(\frac{1280\text{x}720\text{pixels}}{frame}\right)\left(\frac{30\,frames}{\sec}\right)\left(\frac{3color}{frame}\right)\left(\frac{8bits}{color}\right)=663.55\text{Mbps}$$

Above example shows that digital video is rich in terms of data rate required to transmit over any communication channel as well as size of storage memory required. This raw digital video contains vast amount of data, on other side storage and communication capabilities are limited and expensive. Thus, to reduce the size of digital video several compression algorithms had been developed. Human Visual System (HVS) can perceive small difference in the brightness of the digital video, whereas cannot perceive small differences in the color components of the digital video, so compression algorithm takes advantage from limitations of human visual system to remove redundancy of the digital video, to obtain the video compression.

II. MOTION ESTIMATION

Motion Estimation is one of the most time consuming unit in a digital video encoder. It is refers to a inter frame coding. Inter coding is a way of finding similarities between two frames; called the reference frame and the current frame, and

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

then encoding the residual of these two frames, which is the difference of these two frames. Motion estimation exploits temporal redundancy between the video frames, to scope massive visual information compression. In motion estimation each block in a frame is represented by a motion vector that represents the displacement of the block in current frame with respect to reference frame [1]. In short by motion estimation, we mean the estimation of the disarticulation of image pixels from one frame to another in a time sequence.

III. Evaluation Parameters

The estimation parameters are defined for the comparison of the algorithms that indicate the similarity of the predicted frame with the current frame. For finding the best match, different Block Matching Algorithms are used. There are various parameters in [10], of which the most popular and less computationally expensive is Sum of Absolute Difference (SAD) and Mean Absolute Difference (MAD) given by below equations:

$$\mathrm{SAD} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \qquad \mathrm{MAD} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |C_{ij} - R_{ij}|$$

Another one is Peak-Signal-to-Noise-Ratio (PSNR) is defined as below equation, characterizes the motion compensated frame that is created by using motion vectors (MV) and macro blocks from the reference frame.

PSNR depends on Mean Squared Error (MSE), where N is the size of the macroblock; C_{ij} and R_{ij} are the pixels being compared in current macroblock and reference macroblock, respectively.

$$\mathrm{MSE} = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \qquad \mathrm{PSNR} = 10 * \log_{10}\frac{(2^n - 1)^2}{\mathrm{MSE}}$$

Where (2^n -1) is the peak signal value in the image, and n is the number of bits/image sample.

IV. Block Based Motion Estimation Algorithms

Motion estimation is nothing but the progression of finding motion vectors. It is carried out at video encoder part and has a considerable role in video compression process. The aim of the algorithm is to find the residual frame following motion compensation at the same time as keeping the acceptable computational complexity.

Here we introduce some algorithms for block-based motion estimation ranges from very basic Exhaustive search (ES) algorithm to some fast search algorithms, for example Three Step Search (TSS), Advance or New Three Step Search (NTSS), Simple and Efficient Three Step Search (SETSS), Diamond Search (DS) and Adaptive Rood Pattern Search (ARPS).

A. Exhaustive Search

This algorithm also known as Full Search (FS) is the nearly all computationally expensive block-based matching algorithm. Full search algorithm calculates the cost at each possible position inside the search area. As a result it locates the best match block and gives the highest PSNR among any block matching algorithm (BMA). The advantage of this method is that it guarantees perfect matching. The disadvantage to ES is that due to larger search area, it requires more computations. An example is shown in Fig. 1. More the search area is more perfect the matching will be as well as computationally more expensive. For the given example total number of search points for a macroblock is 225.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

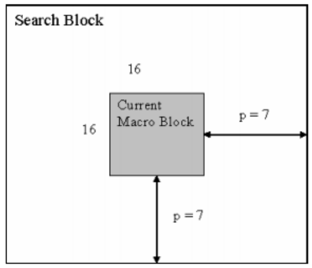

Fig. 1 Macro block of size 16x16 pixels and a search parameter p of Size 7 pixels

B. Three Step Search Algorithm

TSS [2] is one of the earliest attempts at fast block matching algorithm during 1990s. The searching starts from the center location of the reference frame. Initially fixed Step size S = 4 is set. Along with the center location, it searches eight more locations, +/- S pixels from the center pixel as represented in Fig. 2. Each dot in the Fig. 2 represents the center of a macroblock.

The minimum weight at the nine points is found out. That points becomes the center point for the next step and for that step, eight surrounding points with step size S = S/2 is taken and again minimum weight are calculated. This procedure continues until S = 1. The position corresponding to the minimum weight in the final step is taken as the motion vector (MV). In the above illustration, motion vector is (5, -3). The algorithm is applied for each and every macroblock of the current frame taken. Here best match is not guaranteed as in exhaustive algorithm, but the number of computations reduces to 25.

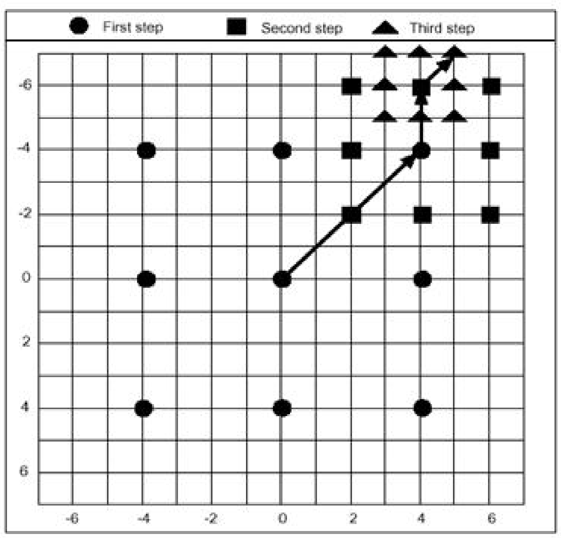

Fig. 2 Three step search procedure

C. New Three Step Search Algorithm

Li et al. proposed the New Three Step Search (NTSS) improves on TSS results by providing a center biased searching scheme and having half way technique to reduce computational cost. The algorithm suits well for searching large motion. Different advancements in three step search procedures are explained in [3] and [4]. In this algorithm, along with the center location, 16 more locations are considered in the first step i.e. 8 locations S = 4 away from the origin and 8 other locations S = 1 away from the origin as shown in Fig. 3. If the minimum is at any of the S = 1 locations, go to step 3, otherwise proceed. Step 2 is keeping the minimum point as center, continue the three step search procedure and Go to step 1.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

Step 3 is go for the neighbouring points either 3 or 5 points depending on the position of the minimum at S = 1 points. If the minimum weight point is at any one of the vertices of S = 1 points, consider 5 neighbouring points and again find the minimum weight and the corresponding position gives the motion vector. Otherwise take the neighbouring 3 points and find the minimum weight and the corresponding position gives the motion vector.

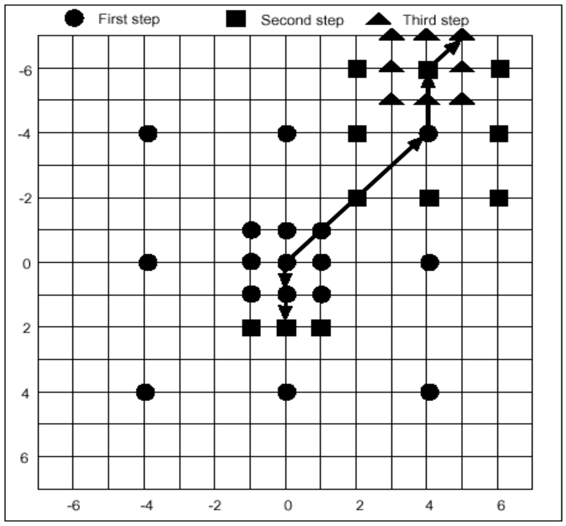

Fig. 3 New three step search procedure

D. Simple and Efficient Three Step Search Algorithm

In this algorithm [5] first consider the 3 points A, B & C as shown in Fig. 4 (a). Point A refers to the center location and B & C are S = 4 away from A towards right hand side and bottom. In the first step along with the 3 search points A, B & C; few more points are considered. For getting those additional points, the following conditions are checked.

If SAD(A) $\geq$ SAD(B) & SAD(A) $\geq$ SAD(C), select (b) in Fig. 5

If SAD(A) $\geq$ SAD(B) & SAD(A) $\leq$ SAD(C), select (c) in Fig. 5

If SAD(A) $<$ SAD(B) & SAD(A) $<$ SAD(C), select (d) in Fig. 5

If SAD(A) $<$ SAD(B) & SAD(A) $\geq$ SAD(C), select (e) in Fig. 5

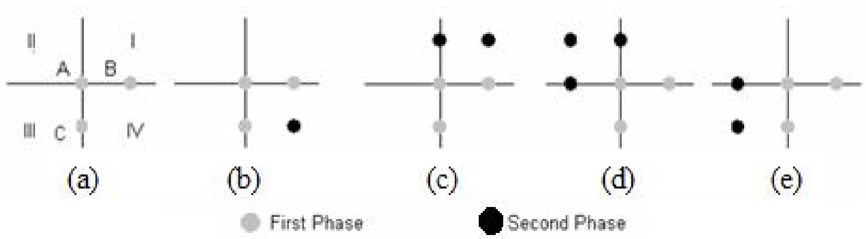

Fig. 4 Search points in each quadrant

Thus from the above conditions and considering Fig. 4, we get the search points for the first step. Find the minimum weight point for the selected points; apply the same procedure again with half step size. The procedure is continued till S = 1.

E. Four Step Search Algorithm

The Four Step Search algorithm (FSS) is proposed by L.M. Po and W. C. Main 1996 [6]. This algorithm also exploits the center-biased characteristics of the real world video sequences by using a smaller initial step size compared with TSS. The initial step size is fourth of the maximum motion displacement d (i.e. d/4). Due to the smaller initial step size,

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

the FSS algorithm needs four searching steps to reach the boundary of a search window with d = 7. Same as the small motion case in the NTSS algorithm, the FSS algorithm also uses a halfway-stop technique in its second and third step's search. Fig. 6 shows two search paths of FSS for searching large motion. For the lower-left path, it requires (9+5+3+8) = 25checking points. For the upper-right path, it requires (9+5+5+8) = 27checking points that is the worst case of the algorithm for d = 7.

Fig. 5((c), (d)) shows two search paths of FSS for searching small motion. For the left path, it requires (9 + 8) = 17 checking points. For the right path, it requires (9+ 3+ 8) = 20 checking points. As shown in Fig. 6 ((a), (b)), there are either three or five checking points required in the second or third searching step. Moreover, if the minimum weight checking point of that searching step is the center one, the step size is reduced by half and jump to the forth step. For the general case, the algorithm can be extended as follows. If the step size of the forth step is greater than one, then another four-step search is performed with the first step equals to the last step of the previous search. The number of checking points required for the worse case is (18 log2 [(d+1)/4] + 9).

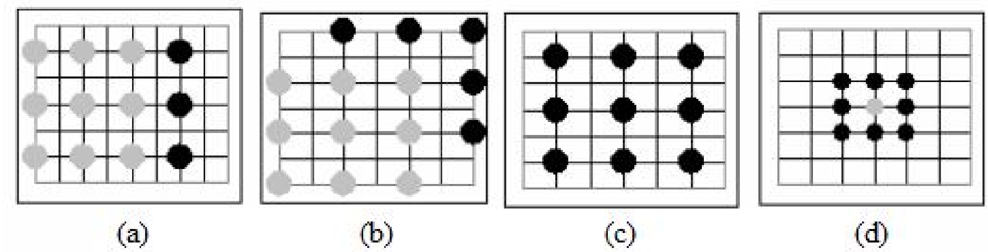

Fig. 5 Search patterns of four step search

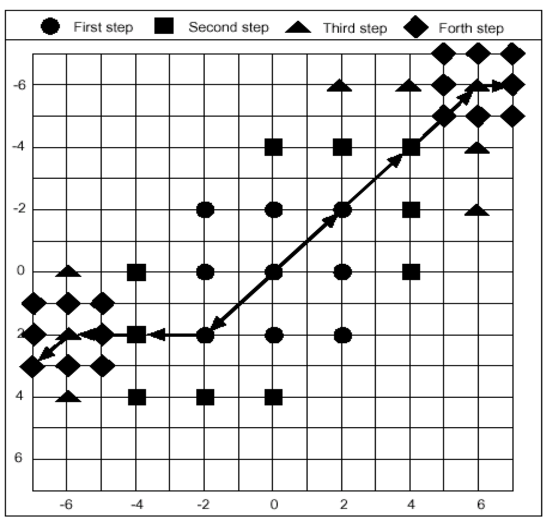

Fig. 6 Four step search procedure

F. Diamond Search Algorithm

The algorithm [7] is almost same as the four step search algorithm except the shape has been changed from square to diamond and there is no limit for the number of steps. Two search points used in this algorithm are Large Diamond Search Procedure (LDSP) and Small Diamond Search Procedure (SDSP). An example for this algorithm is represented in Fig. 7. The motion vector in the example is (-4, -2).

LDSP is applied for the initial steps. In the first step 9 locations are searched and then according to the minimum weight, further steps are continued. Additional locations will be either 3 or 5 depending upon the position of the

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

minimum weight as shown in Fig. 7. The procedure continues until we get the minimum weight at the center position. Then as a last step, SDSP is applied to get the accurate least weight position.

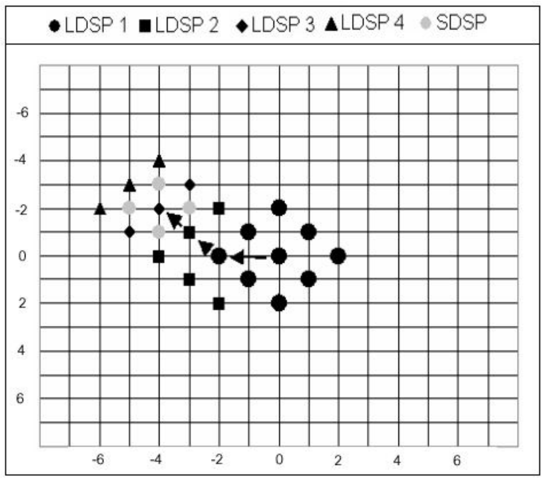

Fig. 7 Diamond search procedure

Since the search pattern is neither big nor small and since there is no limit for the number of steps, matching will be accurate. Matching will be almost equal to exhaustive search while the number of computations will be significantly less.

G. Adaptive Rood Pattern Search Algorithm

This algorithm [8] considers the fact that the motion in most of the portions in a frame will be usually homogeneous. So an adaptive searching algorithm will help a lot for the motion estimation. In this algorithm, for predicting a motion vector of a particular block, motion vector of the immediate left hand side block is taken. Consider the following example.

Assume the motion vector of a block is (3, -2). When we take the very next block, 6 locations will be considered of which the first location will be the center point itself and the second point will be the point taken from the previous MV i.e. (3, -2). Thus the searching is directly put in a point where there is the highest possibility of getting the exact match. Further, the remaining 4 points are obtained by taking a step size, S = Max (|x|, |y|). According to the example it is Max (|3|,|-2|) = 3. Thus for the first step the 6 points are as shown in the Fig. 8. If the minimum is at the center point itself there is no further searching. Otherwise it goes to the second step where SDSP searching is done. It continues until the least weight point is obtained at the center of the SDSP search pattern.

If the predicting vector is far away from the center, the searching is directly put in a point where there is the highest possibility of getting the exact match, so that the number of computations can be saved to a great extend compared to other methods without trading off the best matching guarantee.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

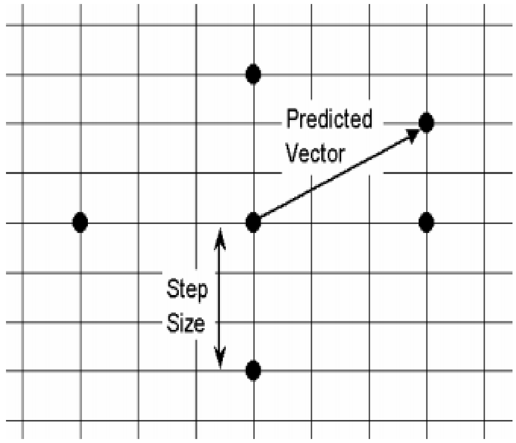

Fig. 8 Adaptive rood pattern: The predicted MV is (3, -2), and the step size S = Max (|3|, |-2|) = 3

V. THE PROPOSED SCHEME

In order to ease the computational burden involved in DS and ARPS for different types of video sequences (both slow movements and fast movements), hybrid motion estimation technique (H-MET) algorithm is proposed which involves less computation overload for any type of video sequences.

The pattern for the proposed algorithm is shown in Fig. 9. It involves the pattern considered for both DS [7] and ARPS algorithms. The performance is evaluated with matching criteria function, sum of absolute difference (SAD) and the experimental results show that the proposed scheme performs better than DS and ARPS [8] scheme in terms of computational time and search point requirements.

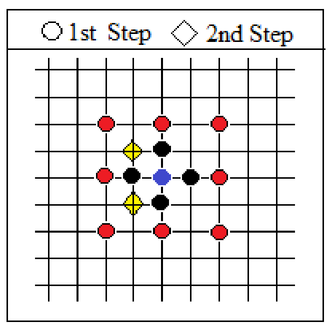

Fig. 9 Cross section pattern for proposed scheme

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

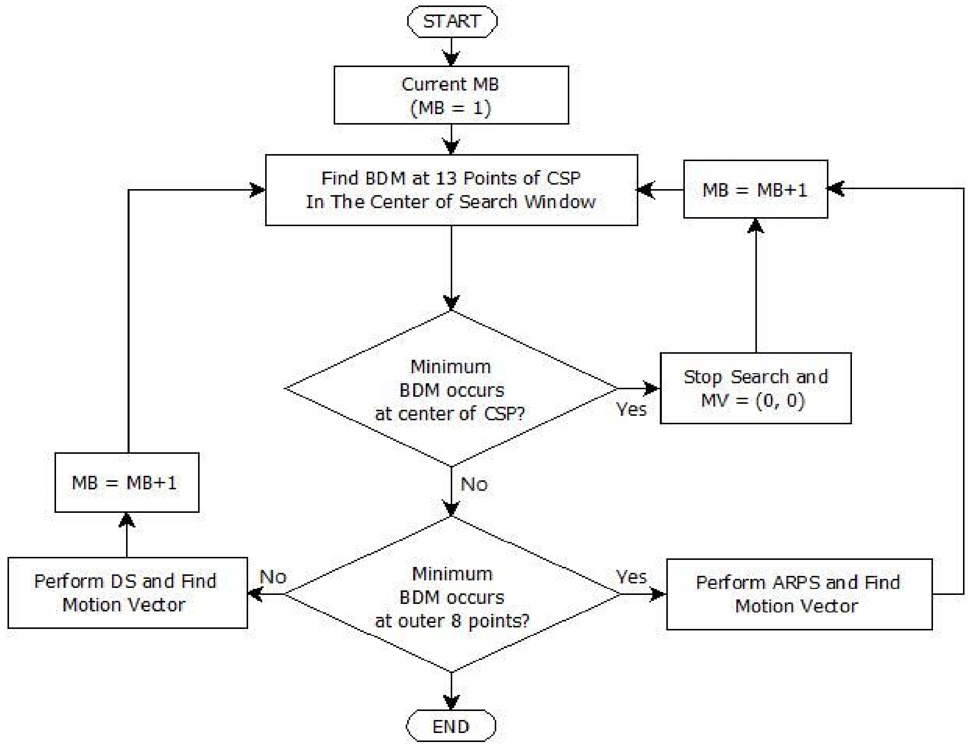

Fig. 10 Flow diagram of proposed hybrid motion estimation technique

The proposed hybrid motion estimation algorithm is recapitulated as follows.

Step-1. The preliminary CSP is center of the search window, and the 13 points of CSP are tested. If the Minimum Block Distortion (MBD) point calculated is to be found at the center position then declare motion vector is (0, 0) and then go to **Step-4**; otherwise, go to **Step-2**.

Step-2. If Minimum Block Distortion (MBD) is found at any of inner four points of CSP then perform diamond search (DS) and calculate motion vector; otherwise, go to **Step-3**.

Step-3. If Minimum Block Distortion (MBD) is found at any of outer eight points of CSP then perform adaptive rood pattern search (ARPS) and calculate motion vector; otherwise go to **Step-4**.

Step-4. The CSP is re-positioned as the center point for new macroblock to form a new motion vector and go to **Step-1**.

Step-5. Repeat this process until the entire frame is scanned.

Fig. 10 shows the detailed flow chart for the implementation of hybrid block-based motion estimation algorithm. The following legends are used: MBD-Minimum Block Distortion; BDM-Block Distortion Measure; CSP-Cross Section Pattern; LDSP-Large Diamond Search Pattern; DS-Diamond Search; MV-Motion Vector; MB-Macro Block

VI. EXPERIMENTAL RESULTS AND DISCUSSIONS

During the course of this project above three algorithms, namely DS, ARPS and hybrid motion estimation technique (H-MET) have been implemented in MATLAB tool. Different standard video sequences were used to generate the frame-by-frame results of the algorithms. They consist of different degrees and types of motion content. Due to space limitation, we only present the one representative among the twenty sets of vigorous simulations. The proposed algorithm is tested with standard video sequences and the results are compared with DS & ARPS algorithms respectively. The first sequence is the "Foreman" in QCIF format. For "foreman" video sequence, the sum absolute difference (SAD) as the BDM, block size of 16, and search parameter of 7 is used.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

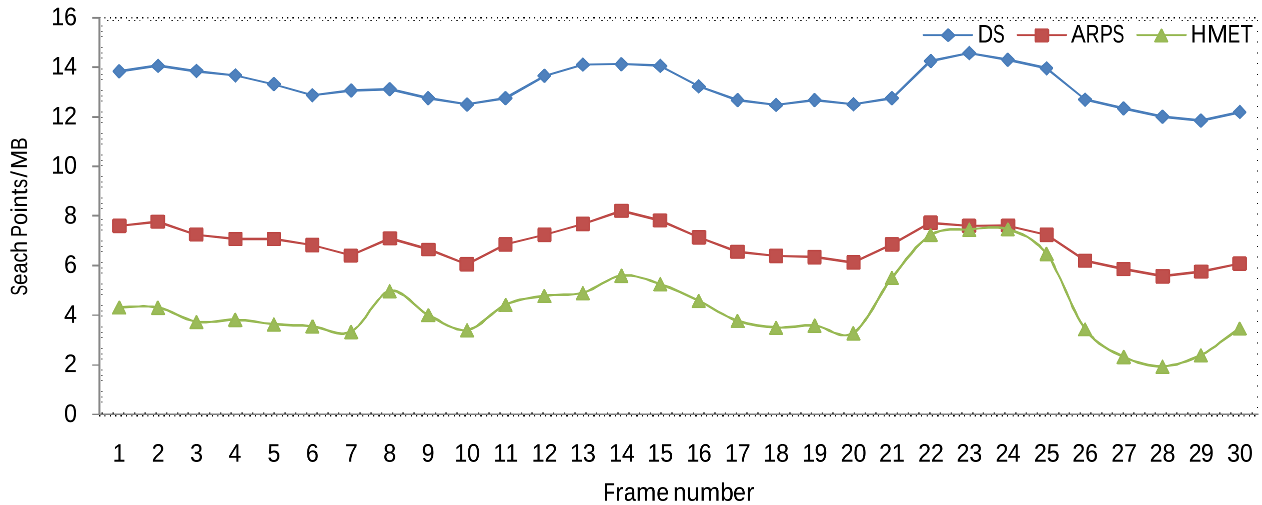

Fig. 11 Computational Complexity for Foreman sequence based on SAD function

It can be seen from the Fig. 11 that the proposed technique requires less search points compared to the DS and ARPS schemes irrespective of the number of frames. Fig. 12 shows that PSNR value of H-MET is obtained for the proposed technique found to be almost identical to the DS and ARPS, the computational complexity along with the fidelity results together suggest the suitability of the proposed technique for real-time video compression problems.

The conclusion of the proposed algorithm is compared with the algorithms adopted by MATLAB tool to verify its effectiveness and efficiency, which is described in Table 2.

MATLAB screen shot of Reference frame (I), current frame (P), and compensated frame (C) and difference between compensated frame and current frame shown in Fig 13.

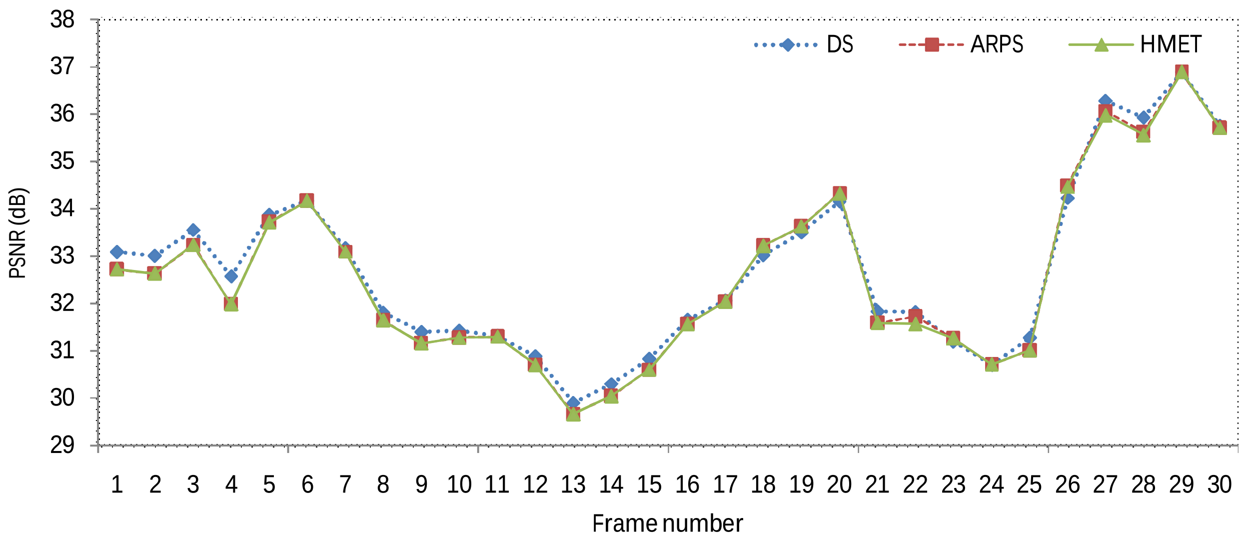

Fig. 12 PSNR performance of fast block matching algorithms

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

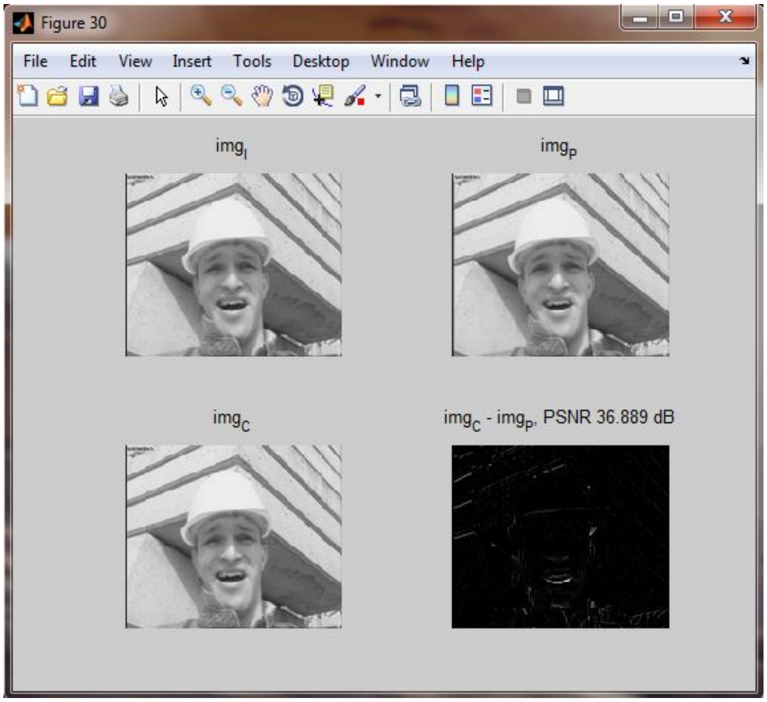

Fig. 13 Reconstructed frame img_c from reference frame img_I and current frame img_p

5 video sequences covering low speed, moderate speed and high speed movement and different temporal details are chosen to evaluate the performance of the proposed technique in the experiment. Experimental parameter settings are as follows: Frames to be tested is beginning 30, Search range parameter is 7.

Video sequence	Algorithm	Avg. Search Points	Avg. PSNR (dB)	Avg. Search Points	Avg. PSNR (dB)
		Macroblock Size 16x16		Macroblock Size 8x8	
Viptraffic	DS	12.76	27.89	14.07	30.31
	ARPS	6.2	27.66	6.48	30.25
	H-MET	1.84	26.84	6.48	30.20
Claire	DS	13.32	41.62	14.75	42.11
	ARPS	6.75	41.57	7.59	42.10
	H-MET	2.79	41.57	4.26	42.09
Carphone	DS	13.88	32.56	15.52	33.27
	ARPS	8.20	32.53	8.69	33.22
	H-MET	6.17	32.52	6.88	33.88
Mother daughter	DS	12.10	38.74	13.68	44.62
	ARPS	5.94	38.74	6.63	44.60
	H-MET	2.42	38.70	2.87	44.60
Akiyo	DS	12.52	44.20	15.11	44.76
	ARPS	5.55	44.09	8.00	44.80
	H-MET	1.20	44.89	4.56	44.80

Table 2. Performance comparison

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

VII. CONCLUSION

In this paper we presented need of motion estimation in video compression and how we can implement the ME techniques. A hybrid motion estimation technique, H-MET, simple yet effective, has been proposed in this work for the digital video compression. The proposed technique performs based on the principle of diamond search (CDS) and adaptive rood pattern search (ARPS) algorithms where the fasts as well as slow motion are successfully detected. The performance of the proposed technique was estimated in terms of the cost function SAD, number of search points and reliability measure PSNR. It has been found from the experimental results that the H-MET scheme yields significant results compared to the DS and ARPS schemes in terms of computational complexity as is shown by table1. H-MET scheme come pretty close to the PSNR results of ARPS with reduced number of computation.

REFERENCES

[1] Borko Furht, Joshua Greenberg, Raymond Westwater, Motion Estimation Algorithms for Video Compression. Massachusetts: Kluwer Academic Publishers, 1997. Ch. 2 & 3.

[2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. National Telecommunications Conf., New Orleans, LA, pp. G5.3.1–G5.3.5., 1981.

[3] Renxiang Li, Bing Zeng, and Ming L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 4., no. 4, pp. 438-442, August 1994.

[4] SUN Ning-ning, FAN Chao, XIA Xu, "An Effective Three-step Search Algorithm for Motion Estimation", 2009 IEEE, 978-1-4244-3929-4/09.

[5] Jianhua Lu, and Ming L. Liou, "A Simple and Efficent Search Algorithm for Block-Matching Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 7, no. 2, pp. 429-433, April 1997

[6] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313-317, 1996.

[7] Shan Zhu, and Kai-Kuang Ma, " A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, vol 9, no. 2, pp. 287-290, February 2000.

[8] Yao Nie, and Kai-Kuang Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, vol 11, no. 12, pp. 1442-1448, December 2002.

[9] Chun-Ho Cheung, and Lai-Man Po, "A Novel Cross-Diamond Search Algorithm for Fast Block Motion Estimation", IEEE Trans. Circuits And Systems For Video Technology, vol 12., no. 12, pp. 1168-1177, December 2002.

[10] Aroh Barjatya, "Block Matching Algorithms For Motion Estimation", Digital Image Processing 6620, pp. 1-6, April 2004.

[11] Pengbin Fu, Haifei Xiong, Huirong Yang, "A Motion Estimation Algorithm for Educational Video Compression", IEEE Trans. Workshop on Digital Media and Digital Content Management, pp. 257-260, 2011.