VOL. 11, NO. 18, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

INVESTIGATION OF MOTION ESTIMATION PERFORMANCE USING HYBRID OF DIAMOND AND ORTHOGONAL SEARCH ALGORITHM

Abdul Majid Darsono, Nurul 'Atiqah Hamid, Nurulfajar Abdul Manap, Redzuan Abdul Manap and Nik Mohd Zarifie Hashim
Centre for Telecommunication Research and Innovation, Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Melaka, Malaysia
E-Mail: abdmajid@utem.edu.my

ABSTRACT

Recent years has shown that many fast Block Matching Algorithm or BMAs have been proposed and developed that can further reduce the computational cost while maintaining the video signal quality. In this project, a new algorithm called Diamond-Orthogonal Search (DOS) is proposed which employs small diamond search pattern and orthogonal shape in its search steps. Additional step to predetermine static block at initial stage is added to further speed up the search which beneficial to small motion video sequence contents. The proposed algorithm and several established algorithms, namely Full Search (FS), Three-Step Search (TSS) and Hexagon-Diamond Search (HDS) are implemented using MATLAB and their performance are being compared and analyzed in terms of peak signal-to-noise ratio (PSNR), and number of search points. Simulation result shows that motion vectors can be find with fewer number of search points while maintaining close video quality performance with other selected algorithm

Keywords: motion estimation, fast block matching algorithm, diamond orthogonal search.

INTRODUCTION

Until today, video compression algorithm has been applied in various video applications ranging from video conferencing to video phones. One of the popular and effective methods in video compression technique is the Motion Estimation (ME) as it reduces the temporal redundancy between successive frames of a video sequence. In ME, motion is estimate by finding the motion vectors of the objects in an image sequence (Ian, Bala and George, 2011). Block matching based ME or best known as Block Matching Algorithm (BMA) is widely used in most of the video codecs, including the H.26x series (Rijkse1996), (Vetrivel, Suba and Athisha, 2010). due to its implementation simplicity and its high compression efficiency (Oliveri, 1997).

The simplest and straightforward BMA is the Full Search (FS) algorithm which exhaustively checking all the possible displacement within the search window to find the best matching block. However, due to its high computational complexity, it is not a best choice for real-time video coding implementation.

Since then, many new and various type of fast BMAs are developed and proposed in order to reduce the computational cost as much as possible while maintaining the accuracy degradation of ME. Those are, for example, the Three-Step Search (TSS) (Koga, 1981), New-Three-Step Search (NTSS) (Li, Zeng and Liou, 1994), Four-Step Search (4SS) (Po and Ma, 1996), Hexagon-Diamond Search (HDS) (Manap *et al.*, 2010), Diamond Search (DS) (Zhu and Ma, 2000), Cross Diamond Hexagonal Search (CDHS) (Cheung and Po, 2005), Cross Diamond Search (CDS) (Jia and Zhang, 2004) and Orthogonal Search (OS) (Soongsathitanon, Woo and Dlay, 2005).

In the real-time video sequences, the distribution of motion vector (MV) is highly center-biased which leads to development of center biased BMAs. This eventually provides the close prediction accuracy, especially for slow motion video sequences. For video sequences with large motion, TSS and OS are more efficient to find the global minimum but tends to be trapped into local minimum for small motion video sequences (Soongsathitanon, Woo and Dlay, 2005), (Ian, Bala, and Anitha, 2011).

For this project, Diamond - Orthogonal Search (DOS) is proposed and implemented using several standard video sequences in MATLAB. A few established BMA algorithms, namely FS, TSS and HDS are being compared with the proposed algorithm. Their performances are then compared and analyzed in terms of peak signal-to-noise ratio (PSNR), number of search points needed as well as their computational complexity in order to determine their suitability to different motion content represented in those video sequences.

BLOCK MATCHING ALGORITHM

BMA has been widely adopted by current video coding standards such as H.26x and MPEG series due to its effectiveness and simplicity for implementation of ME. BMA also best described as the technique that estimates the amount of motion on a block by block basis. Best block from the previous frame is search to reconstruct an area of the current frame by dividing each coding frame into non-overlapping blocks with size of M-by-N pixels. The macro blocks are then compared with corresponding block and its adjacent neighbors in the previous frame within a search area of size ($M + 2q$ x $N + 2q$). The general idea is shown in Figure 1 where q is the maximum displacement allowed, A is the search window in the previous frame, B is the block in the current frame and C is the block in the previous frame (Srinivasan and Rao, 1985). Motion is defined as a displacement of an object over a period of time and measured in two consecutive frames of a video image. Therefore, motion vector (MV) is best defined as the displacement between the current block

VOL. 11, NO. 18, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

and previous block which is found by the best match based on certain matching cost function (Takaya, 2006).

Based on (Takaya, 2006), usually the macro block is taken as M = 16 pixels and N = 16 pixels and the search parameter q = 7 pixels. The larger motion requires larger q and the larger the search parameter the more computationally expensive the process of ME becomes.

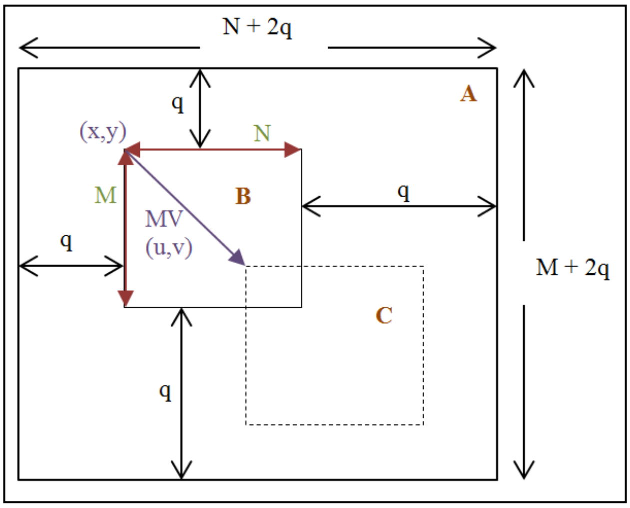

Figure-1. Current and previous frames (M x N) in a search window (N+2q x M+2q) and the motion vector (MV).

Matching cost function

Matching Cost Function Is Defined To Measure The Similarity Betwwen The Candidate Macroblock (Mb) And Current Mb. There Are Several Matching Cost Functions That Can Be Used For This Purpose, Such As Mean Squared Error (Mse), Sum Of Absolute Difference (Sad) (Metkar and Talbar, 2013), Mean Squared Difference (Msd) And Mean Absolute Different (Mad) (Usama, Montaser And Ahmed, 2005).

One Of The Major Factors Affecting The Me Algorithm Complexity And Its Performance Is The Cost Function. Sad And Msd Are The Two Well-Known Cost Functions Due To Its Simplicity Where The Distortion Or The Matching Error Between The Block Must Be Minimized To Obtain The Best Match. The Functions Are Briefly Explained As Follows:

Sum of Absolute Difference (SAD)

$$SAD(i,j)=\sum_k\sum_l\left|C_f(k,l)-R_f(k+i,l+j)\right| \quad (1)$$

Mean Squared Difference (MSD)

$$MSD(i,j)=\frac{1}{MN}\sum_k\sum_l\left[C_f(k,l)-R_f(k+i,l+j)\right]^2 \quad (2)$$

In the measure, $C_f\ (k,l)$ is the location of the pel at the uppermost left in the block of the current frame, f while $R_{f-1}(k+i,l+j)$ is the location of the pel on the previous frame f-1, shifted by the (i,j) within the search area. Meanwhile, MN is the block size and the smallest $MSD(i,j)$ or $SAD(i,j)$ within the search area represents the best match.

Peak signal-to-noise ratio

Another metric is the Peak Signal-to-Noise ratio (PSNR). It is used to determine the quality of the compressed images. The higher the PSNR value, the better the quality of the compensated images (Metkar and Talbar, 2013). The PSNR equation for a grayscale image is defined as follows (Barjatya, 2004):

$$PSNR = 10\log_{10}\left(\frac{(\text{Peak to peak value of original data})^2}{MSD}\right) = 10\log_{10}\left(\frac{255^2}{\frac{1}{MxN}\sum_{i=1}^{M}\sum_{j=1}^{N}(x_{ij}-\hat{x}_{ij})^2}\right) \quad (3)$$

where (MN) is the dimension of the frame in pixels while x_{ij} and $\hat{x}_{ij}$ are the luminance components of the original and the reconstructed image, respectively, at the spatial location (i,j).

PROPOSED ALGORITHM

In this search algorithm, Diamond - Orthogonal Search Algorithm (DOS) employs two different patterns which combine Small Diamond Search Pattern (SDSP) and Orthogonal shape for its search steps.

Based on (Nie and Ma, 2002), a large percentage of zero-motion blocks occurred in many visual communication applications such as video telephony. Therefore, additional step to determine whether the block is static is added at the beginning of the search with a predetermined threshold T.

It is stated in (Nie and Ma, 2002), that the average SAD of the static macroblocks (MBs) is within the range of 600 and 1300. Therefore, value T=512 is fairly chosen to increase search without causing noticeable degradation on visual quality.

The block matching is conducted within the 15x15 search window size and the block size is fixed at 16x16 and the frame distance between predicted frame and original frame is set to be 1 for consistent comparison with previous research works. The maximum displacement allowed is ±7 horizontally and vertically. The matching cost function used is SAD in the procedure which adds up the absolute differences between corresponding element in the current and reference blocks (Jia and Zhang, 2004).

The proposed DOS algorithm steps are summarized as follows:

Step 1: The matching error (SAD_{center}) between current block and the block at the same location in the

VOL. 11, NO. 18, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

reference frame is computed with threshold equal to 512. If the SAD_{center} less than 512, the final MV is [0 0] and the search is stop. Otherwise, proceed to step 2.

Step 2: The first five points initiate the small diamond search pattern (SDSP) where the center point is centered at the origin of the search window with step size of 2, as shown in Figure-2. All points are tested to find the minimum cost function (MCF) point. If the MCF point is found at the center, proceed to Step 3. Otherwise, the orthogonal pattern (horizontal) with step size of 1 is employed to the center point and proceeds to Step 3.

Step 3: Two vertical search points with step size of 2 are searched for MCF point and proceed to Step 4.

Step 4: The step size is reduced to 1 point for the two horizontal search points and then proceeds to Step 5.

The search pattern and step size is the same as Step 4 but in vertical direction and the MV is found in this step is the final MV.

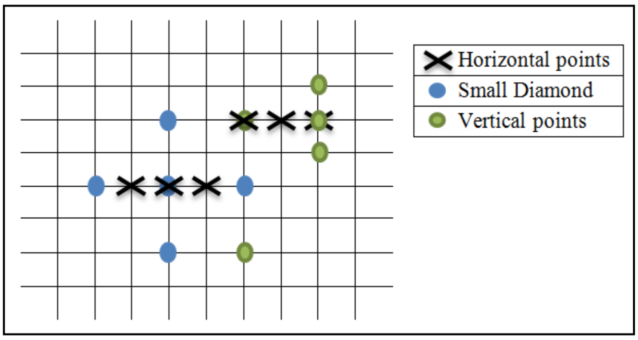

Figure-2. The proposed orthogonal diamond search pattern.

RESULTS AND ANALYSIS

Type of classes for test video sequence

Three different video sequences of standard Quarter Common Intermediate File Format (QCIF) are chosen with different types of motion contents. Video sequences are differentiated according to their types of motion complexity namely Type I, Type II and Type III as shown in Table 1[16].

Table-1. Video sequences according to their respective classes.

Type	Video Sequence	Content motion
I	Silent, Claire, Akiyo	Slow/low
II	News, Suzie, Coastguard	Moderate
III	Foreman, Tennis, Football	High/complex

Performance analysis

The first 50 frames of total frames of the video sequences are considered to be simulated and analyzed for this algorithm. The video sequences used for this analysis are Akiyo, Tennis and Coastguard representing each type of motion contents. The original frame rate is 15 frames per second (or fps).

The performance is compared between three established algorithms, namely Full Search (FS), Three Step Search (TSS), Hexagon-Diamond Search (HDS) with the proposed algorithms Diamond Orthogonal Search (DOS) algorithm. The results are shown in Table-2 and Table-3. A few graphs or figures for the simulations are shown later for further insight.

Table-2. Average number of search pattern per block frame for fs, tss, hds and dos.

Video	Algorithm			
	FS	TSS	HDS	DOS
Akiyo	225	25	11.00	1.70
News	255	25	11.04	2.22
Coastguard	255	25	11	4.97

Table-3. Average psnr per block frame for fs, tss, hds and dos.

Video	Algorithm			
	FS	TSS	HDS	DOS
Akiyo	43.83	43.51	43.51	43.51
News	37.74	37.17	37.17	37.17
Coastguard	31.63	28.34	28.34	28.34

For comparison, the performance of FS, TSS, HDS and DOS are documented and reported as below. The average number of search pattern per block frame is tabled as shown in Table-2 for easier understanding. Meanwhile, the average Peak Signal-to-Noise ratio (PSNR) per block frame of each reconstructed video sequence computed for quality comparison is tabled in Table-3.

Based on Table-2, DOS algorithm greatly improves and contributes lesser search points compared to FS, TSS and HDS algorithm. The highest average search point is from the FS algorithm as it searches all possible search point within the search window in finding the optimum minimum points. Furthermore, the addition step at the initial stage helps in improving the search steps as it avoids the local minimum matching cost error points. Therefore, it can be said that the average search points per block are DOS < HDS < TSS < FS respectively.

As for Table-3, it can be seen that the FS gives the highest PSNR value compared to the other three which give similar PSNR values. There is a slight degradation in quality for moderate and fast motion sequences compared to the small motion sequence.

VOL. 11, NO. 18, SEPTEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

The average search points and PSNR performances are plotted on frame-by-frame for “Akiyo”, “News” and “Coastguard” video sequences representing all the three classes of motion contents.

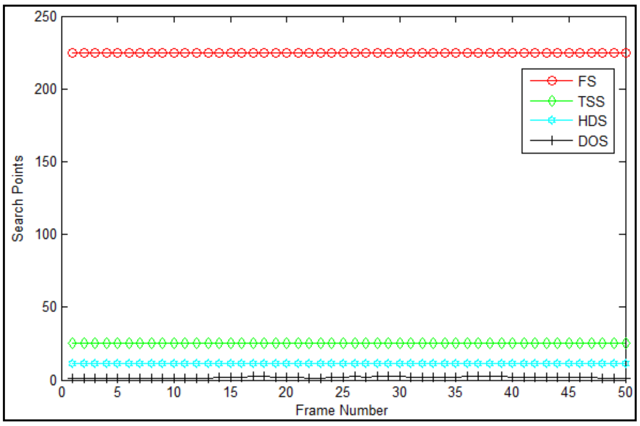

Figure-3. Comparative average search points per block per frame for "Akiyo" sequence (Type I).

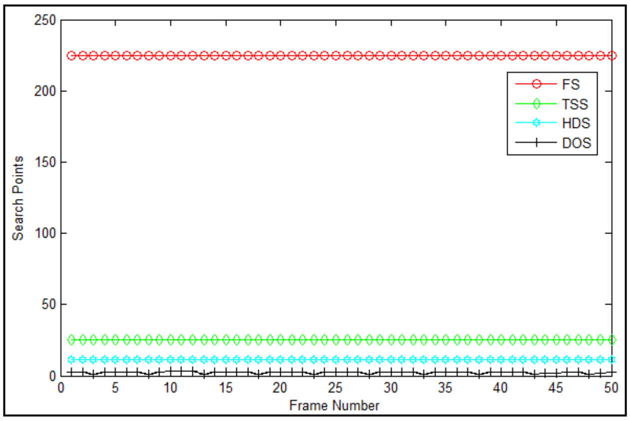

Figure-4. Comparative average search points per block per frame for "News" sequence (Type II).

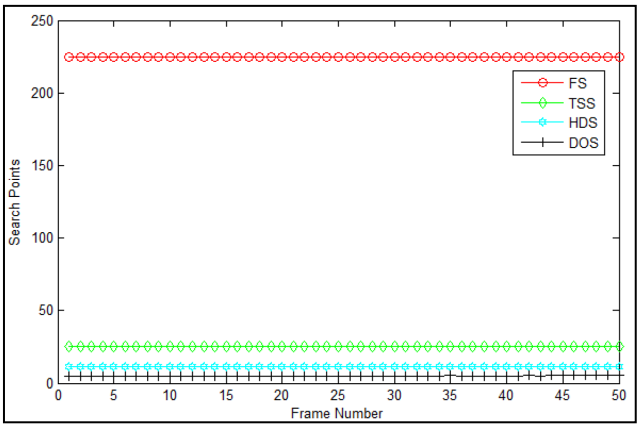

Figure-5. Comparative average search points per block per frame for "Coastguard" sequence (Type III).

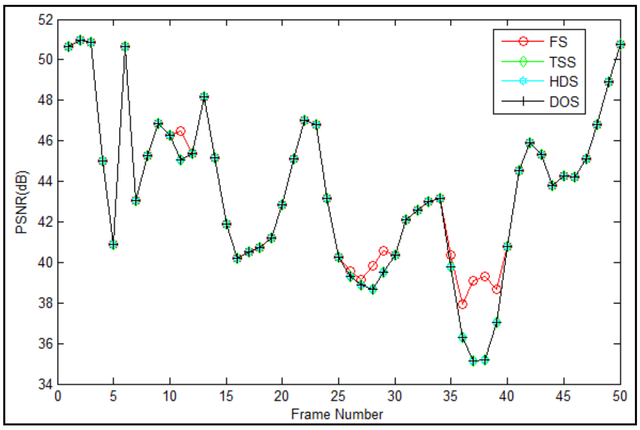

Figure-6. Comparative average PSNR per block per frame for "Akiyo" sequence (Type I).

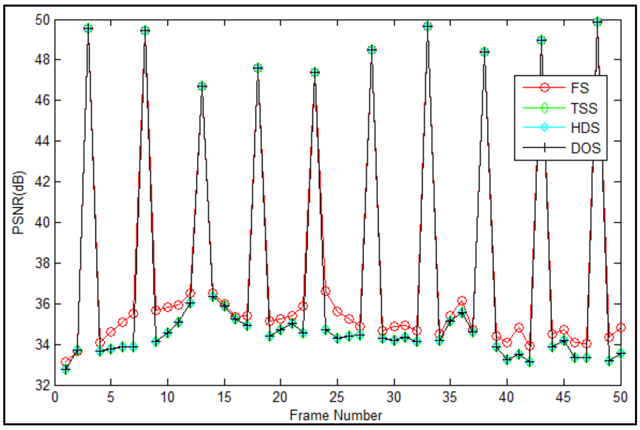

Figure-7. Comparative average PSNR per block per frame for "News" sequence (Type II).

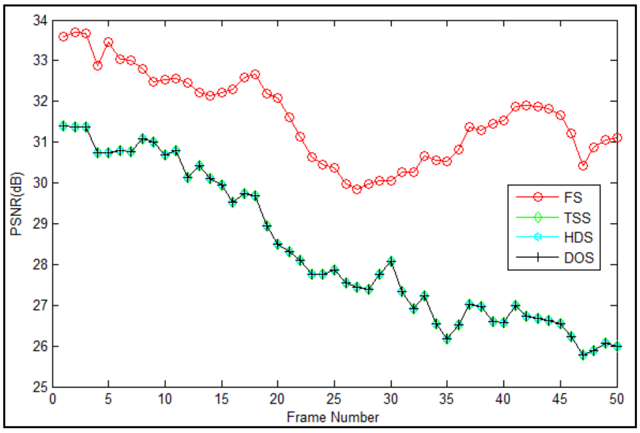

Figure-8. Comparative average PSNR per block per frame for "Coastguard" sequence (Type III).

As for all the above figures, it can be seen that based on Figure-3, Figure-4 and Figure-5 shows that the proposed algorithm, DOS gives the best result where most of the search points are below 10 point max for all motion types. However, similar and close PSNR performance results can be seen in Figure-6, Figure-7 and Figure-8 for all four algorithms.

CONCLUSIONS

A new and simple BMA called Diamond Orthogonal Search Algorithm (DOS) is proposed for the fast BMA motion estimation. It employs two different search pattern and strategy which combines the orthogonal shape and small diamond search pattern. Furthermore, additional step is added to predetermine the static block thus avoiding the extra search step. Result show that DOS performs better in terms of search points compared to the other algorithms for all different types of motion contents. However, in terms of PSNR performance, DOS maintains a close performance when comparing between the other BMAs algorithms.

ACKNOWLEDGEMENT

The authors would like to thank Universiti Teknikal Malaysia Melaka (UTeM) and Ministry of Education Malaysia for their sponsorship through the Exploratory Research Grant Scheme (ERGS/1/2013/TK02/UTEM/03/04/E00020) that makes this research work possible.

REFERENCES

Barjatya, A. 2004. Block matching algorithms for motion estimation. IEEE Transactions Evolution Computation. Vol. 8, iss 3, pp. 225-239.

Cheung, C. and Po, L. 2005. Novel cross-diamond-hexagonal search algorithms for fast block motion estimation. IEEE Transactions on Multimedia, vol 7, iss 1, pp. 16-22, 2005.

Ian, P., Bala, G. and Anitha, J. 2011. Enhanced modified orthogonal search for motion estimation. 2011 IEEE Recent Advances in Intelligent Computational Systems (RAICS), pp. 907-910.

Ian, P., Bala, G. and George, B. 2011. A Study on Block Matching Algorithms for Motion Estimation. International Journal on Computer Science and Eng. Vol. 3, issue 1.

Jia, H. and Zhang, L. 2004. A new cross diamond search algorithm for block motion estimation. Proc. IEEE International Conf. on Acoustics, Speech and Signal Processing, vol. 3, p. 357.

Koga, T. 1981. Motion-compensated interframe coding for video conferencing. Proc. NTC81, pp. 5-13.

Li, R., Zeng, B., and Liou, M. 1994. A new three-step search algorithm for block motion estimation. Circuits and Systems for Video Technology, IEEE Transactions on, vol 4, iss 4, pp. 438-442.

Manap, R., Ranjit, S., Basari, A., and Ahmad, B. 2010. Performance Analysis of Hexagon-Diamond Search Algorithm for Motion Estimation. Proc. International Conf. on Computer Engineering and Technology, vol 3, pp. 3-155.

Nie, Y. and Ma, K. 2002. Adaptive rood pattern search for fast Block Matching Motion Estimation. IEEE Transactions on Image Processing. vol. 11, iss 12.

Oliveria, J. 1997. A Java H. 263 Decoder Implementation. Electrical and Computer Engineering Department, University of Ottawa.

Po, L., and Ma, W. 1996. A novel four-step search algorithm for fast block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology, vol 6, iss 3, pp. 313-317.

Rijkse, K. 1996. H. 263: video coding for low-bit-rate communication. IEEE Communications Magazine, vol. 34, iss 12, pp. 42-45.

Soongsathitanon, S., Woo, W. and Dlay, S. 2005. 'Fast search algorithms for video coding using orthogonal logarithmic search algorithm', IEEE Transactions on Consumer Electronics. Vol. 51, iss 2, pp. 552-559.

Srinivasan, R. and Rao, K. 1985. Predictive coding based on efficient motion estimation. IEEE Transactions on Communications, vol. 33, iss 8, pp. 888-896.

Takaya, K. 2006. Detection of Moving Object in Video Scene - MPEG like Motion Vector vs. Optical Flow'. The First International Workshop on Video Processing for Security (VP4S-06) - Held Jointly with CRV06, Quebec.

Metkar, S. and Talbar, S. 2013. Motion Estimation Techniques for Digital Video Coding, 1st ed. New Delhi: Springer, pp. 13-31, 53-59.

Usama, S., Montaser, M. and Ahmed, O. 2005. A complexity and quality evaluation of block based motion estimation algorithms. Acta Polytechnica, vol. 45, issue 1.

Vetrivel, S., Suba, K., and Athisha, G. 2010. An overview of h. 26x series and its applications. International Journal of Engineering Science and Technology, vol 2, iss 9, pp. 4622-4631.

Zhu, S. and Ma, K. 2000. A new diamond search algorithm for fast block-matching motion estimation. IEEE Transactions on Image Processing, vol. 9, iss 2, pp. 287-290.