VOL. 10, NO. 4, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

A NEW DIAMOND – ORTHOGONAL SEARCH ALGORITHM FOR FAST BLOCK MATCHING MOTION ESTIMATION

Nurul 'Atiqah Hamid, Abdul Majid Darson, Nurulfajar Abdul Manap and Redzuan Abdul Manap
Centre for Telecommunication Research and Innovation (CeTRI), Faculty of Electronics and Computer Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Durian Tunggal, Melaka, Malaysia
E-Mail: m021310019@student.utem.edu.my

ABSTRACT

In video processing, Block Matching Algorithm or BMAs plays an important part as it is widely used in most of well-known video codes due to its simplicity and high compression efficiency. This paper proposed a new algorithm, namely Diamond-Orthogonal Search Algorithm (DOS) which employs large diamond search pattern and orthogonal shape for its search steps. At initial step, an additional step is added to predetermine static block to further speed up the search process as it is beneficial to small motion video sequence contents. Several established algorithms, namely Full Search (FS), Three Step Search (TSS), Hexagon – Diamond Search (HDS) and Orthogonal Diamond Search (ODS) together with proposed algorithm 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. Based on the simulation result, motion vector can be found with fewer search points while maintaining close video quality performance with other algorithms.

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

INTRODUCTION

Up until today, video compression algorithm has been applied in various video applications ranging from video conferencing to video telephony. Motion Estimation (ME) can consider as one of the popular and effective technique in video compression as it reduces the temporal redundancy between successive frames of a video sequence. Motion is estimated by finding the motion vectors of the objects in an image sequence (Pandian *et al.* 2011). In ME, 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 (Rijkse 1996; Vetrivel *et al.* 2010) due to its implementation simplicity and its high compression efficiency (Oliveira 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 good choice for real-time video coding implementation (Sun *et al.* 2009).

Since then, different approaches of fast BMAs have been developed and proposed in order to reduce the computational cost as much as possible while maintaining the accuracy degradation of ME. For example, those are the Three-Step Search (TSS) (Sun *et al.* 2009), New-Three-Step Search (NTSS) (Li *et al.* 1994), Four-Step Search (4SS) (Po & Ma 1996), Hexagon-Diamond Search (HDS) (Manap *et al.* 2010), Diamond Search (DS) (Zhu & Ma 2000), Cross Diamond Hexagonal Search (CDHS) (Cheung & Po 2005), Cross Diamond Search (CDS) (Jia & Li 2004), Orthogonal Search (OS) (Soongsathitanon *et al.* 2005) and Orthogonal Diamond Search (ODS) (Hamid *et al.* 2014).

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 tend to be trapped in local minima for small motion video sequences (Soongsathitanon *et al.* 2005; Immanuel Alex Pandian *et al.* 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, HDS and ODS are being compared with the proposed algorithm. Their performances are then compared and evaluated in terms of peak signal-to-noise ratio (PSNR), computational complexity as well as their number of search points needed 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. The 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 searched 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 + 2p x N + 2p). 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 (Rao & Srinivasan 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

VOL. 10, NO. 4, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

best defined as the displacement between the current block 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 p = 7 pixels. The larger motion requires larger p and the larger the search parameter the more computationally expensive the process of ME becomes.

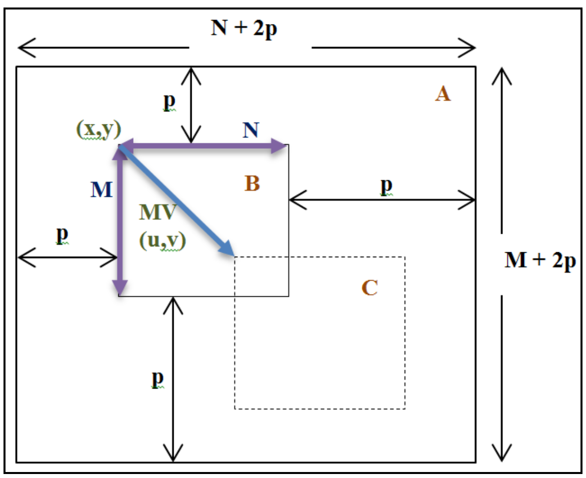

Figure-1. Current and previous frames (N x M) in a Search Window (M+2q x N+2q) and the Motion Vector (MV).

COMPARISON CRITERIA

Matching cost function

Block Distortion Measure (BDM) or Matching Cost Function (MCF) is used to measure the similarity between the candidate macroblock (MB) and the current MB. Several MCF such as mean squared error (MSE) (Metkar & Talbar 2013), sum of absolute difference (SAD) (Metkar & Talbar 2013), mean squared difference (MSD) (Usama et al. 2005) and mean absolute different (MAD) (Usama et al. 2005) can be used for this purpose.

In this paper, SAD and MSD are chosen 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 :

Mean Squared Difference (MSD)

$$MSD(i,j) = \frac{1}{MN}\sum_k \sum_l [C_f(k,l) - R_{f-1}(k+i, l+j)]^2 \qquad (1)$$

Sum of Absolute Difference (SAD)

$$SAD(i,j) = \sum_k \sum_l \left| C_f(k,l) - R_{f-1}(k+i, l+j) \right| \qquad (2)$$

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

The Peak Signal-to-Noise ratio (PSNR) is used to determine the quality of the compressed images. The higher the PSNR value, the better the quality of the compensated images (Metkar & Talbar 2013). The equation of the PSNR is shown as below (Barjatya 2004) :

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

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).

METHODOLOGY

There are four different blocks matching algorithms selected to be tested together with the proposed algorithm. Full Search (FS), Three Step Search (TSS), Hexagon-Diamond Search (HDS) and Orthogonal-Diamond Search (ODS) covers different search strategies and patterns thus make them suitable for comparison.

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

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

Diamond orthogonal search algorithm

The proposed algorithm employs two different patterns which make up the Diamond Orthogonal Search Algorithm (DOS). The two patterns are Large Diamond Search Pattern (LDSP) and Orthogonal search pattern for its search steps.

The proposed DOS algorithm steps are summarized as follows:

VOL. 10, NO. 4, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

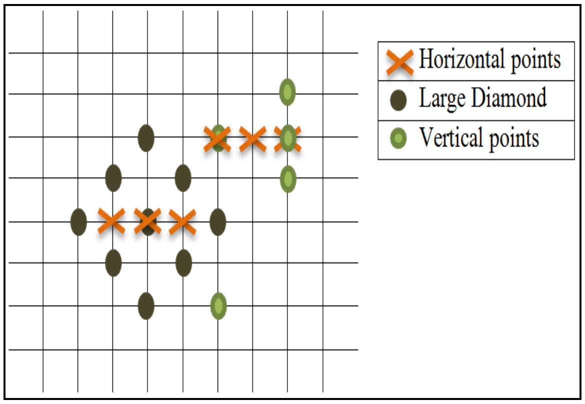

Figure-2. The proposed diamond orthogonal search pattern.

Step 1: The matching error (SADcenter) between current block and the block at the same location in the reference frame is computed with threshold equal to 512. The final MV is [0 0], if the SADcenter less than 512, and the search is stopped. Otherwise, proceed to step 2.
Step 2: The large diamond search pattern (LDSP) is initiated by the first nine points 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 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.
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.

RESULTS AND DISCUSSIONS

All the algorithms are being simulated using MATLAB software. The video sequences used for this purposed are Akiyo, News and Foreman sequences with each representing different types of motion. Each video sequence is representing the Slow/Low motion, Moderate motion and High/Complex motion respectively (Hamid et al. 2014).

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 & Li 2004).

For simulation, only the first 50 frames of the video sequences are chosen to be analyzed. The original frame rate is 15 frames per second (or fps). The results are shown in Table-1 and Table-2. A few graphs or figures for the simulations are shown later for further insight.

Table-1. Average number of search pattern per block frame.

Video	Algorithm				
	FS	TSS	HDS	ODS	DOS
Akiyo	225	25	11.00	10.00	1.77
News	255	25	11.03	10.02	2.35
Foreman	255	25	11	10.00	5.46

Table-2. Average PSNR per block frame.

Video	Algorithm				
	FS	TSS	HDS	ODS	DOS
Akiyo	43.83	43.51	43.51	43.51	43.51
News	37.74	37.17	37.17	37.17	37.17
Foreman	31.63	28.43	28.43	28.43	28.43

The performance of the selected and proposed algorithms is compared and shown in both tables and graphs. The average number of search pattern per block frame is tabled as shown in Table-1 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-2.

After going through the results, as shown in Table-1, some algorithms stand out before the others. The algorithm with the lowest average number of search points is the proposed algorithm with 1.77, 2.35 and 5.46 for Akiyo, News and Foreman respectively. FS is the highest with 225 as it it searches all possible search points within the search window to 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.

Also in Table-2, FS gives the highest PSNR value compared to other algorithms. It can be seen that the other algorithm achieves similar PSNR values.

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

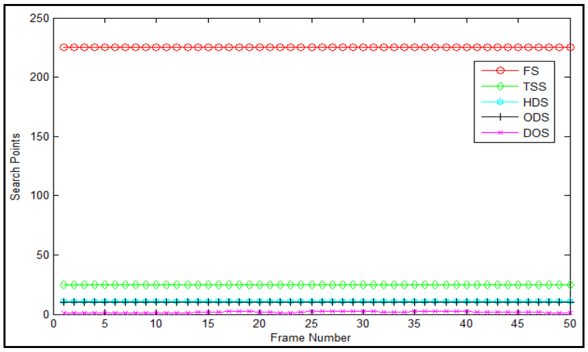

Figure-3. Comparative average search points per block per frame for "Akiyo" sequence (Slow/Low motion).

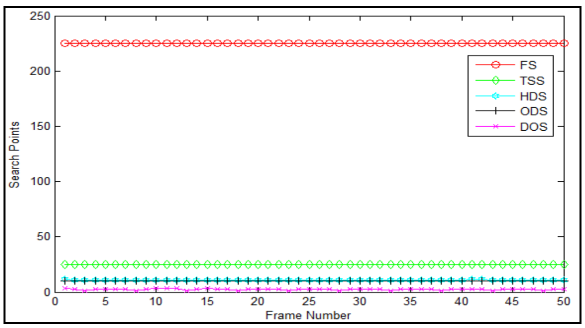

Figure-4. Comparative average search points per block per frame for "News" sequence (Moderate motion).

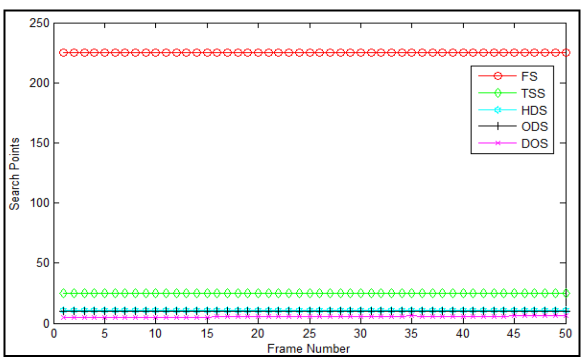

Figure-5. Comparative average search points per block per frame for "Foreman" sequence (High/Complex motion).

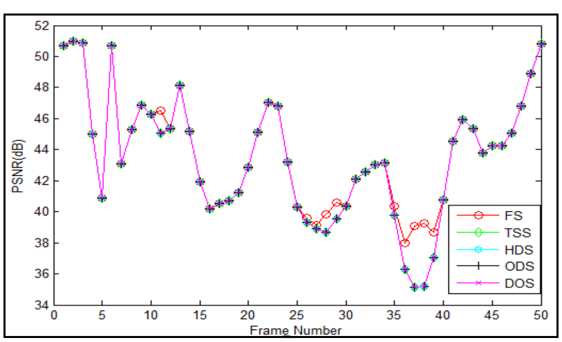

Figure-6. Comparative average PSNR per block per frame for "Akiyo" sequence (Slow/Low motion).

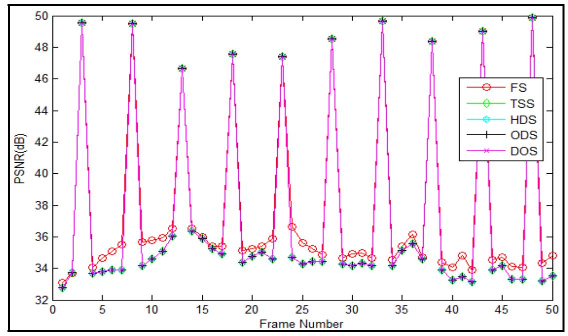

Figure-7. Comparative average PSNR per block per frame for "News" sequence (Moderate motion).

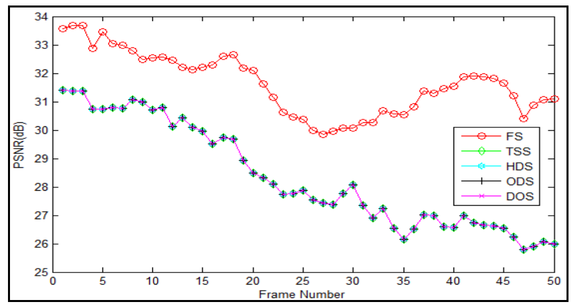

Figure-8. Comparative average PSNR per block per frame for "Foreman" sequence (High/Complex motion).

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, an 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.

ACKNOWLEDGEMENTS

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

[1] Barjatya A., 2004. Block Matching Algorithms For Motion Estimation.

[2] Cheung C. and Po L. 2005. Novel cross-diamond-hexagonal search algorithms for fast block motion estimation. IEEE Transactions on Multimedia. 7(1), pp.16–22.

[3] Hamid N.A. *et al.*2014. A New Orthogonal – Diamond Search Algorithm for Motion Estimation. In International Conference on Computer, Communications, and Control Technology. (I4CT) 2014. pp. 467–471.

[4] Immanuel Alex Pandian S., Josemin Bala G. and Anitha J. 2011. Enhanced modified orthogonal search for motion estimation. 2011 IEEE Recent Advances in Intelligent Computational Systems. pp.907–910.

[5] Jia H. and Li Z. 2004. A New Cross Diamond Search Algorithm For Block Motion Estimation. In IEEE International Conference on Acoustics Speech and Signal Processing. pp. iii – 357–60.

[6] Li R. *et al.* 1994. A New Three-Step Search Algorithm for Block Motion Estimation. IEEE Transactions on Circuits and Systems for Video Technology. 4(4).

[7] Manap R.A. *et al.* 2010. Performance Analysis of Hexagon-Diamond Search Algorithm for Motion Estimation. 2010 2nd International Conference on Computer Engineering and Technology. pp.V3–155–V3–159.

[8] Metkar S. and Talbar S. 2013. Motion Estimation Techniques for Digital Video Coding, Springer India.

[9] Nie Y. and Ma K.-K. 2002. Adaptive rood pattern search for fast block-matching motion estimation. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 11(12), pp.1442–9.

[10] Oliveira J.C. De. 1997. A Java H . 263 Decoder Implementation. University of Ottawa.

[11] Pandian S.I.A., George B.A. and Josemin Bala G. 2011. A Study on Block Matching Algorithms for Motion Estimation. International Journal on Computer Science & Engineering 3(1), pp.34–44.

[12] Po L. and Ma W.-C. 1996. A novel four-step search algorithm for fast block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology. 6(3), pp.313–317.

[13] Rao R. and Srinivasan R. 1985. Predictive Coding Based on Efficient Motion Estimation. IEEE Transaction on Communications. 33(8), pp.888–896.

[14] Rijkse K. 1996. H.263: video coding for low-bit-rate communication. IEEE Communications Magazine. 34(12), pp.42 – 45.

[15] Soongsathitanon S., Woo W. L. and Dlay S.S. 2005. Fast search algorithms for video coding using orthogonal logarithmic search algorithm. IEEE Transactions on Consumer Electronics. 51(2), pp.552–559.

[16] Sun N., Fan C. and Xia X. 2009. An effective three-step search algorithm for motion estimation. 2009 IEEE International Symposium on IT in Medicine & Education pp.400–403.

[17] Takaya K., 2006. Detection of Moving Objects in Video Scene – MPEG like Motion Vector vs . Optical Flow. In The First International Workshop on Video Processing for Security. Quebec City.

[18] Usama S., Montaser M. and Ahmed O. 2005. A Complexity and Quality Evaluation of Block Based Motion Estimation Algorithms. Acta Polytechnica. 45(1).

[19] Vetrivel S. K.Suba and G.Athisha 2010. An Overview of H . 26x Series and its Applications. International Journal of Engineering Science and Technology, 2(9), pp.4622–4631.

[20] Zhu S. and Ma K.-K. 2000. A new diamond search algorithm for fast block-matching motion estimation - Image Processing, IEEE Transactions on. IEEE Transactions on Image Processing. 9(2), pp.287–290.