Adaptive Diamond Search Algorithm for Motion Estimation

Abd Majid Darsono, Nurul Atiqah Hamid, Nurulfajar Abd Manap, Redzuan Abd Manap, Nik Mohd Zarifie Hashim
Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka
Melaka, Malaysia
abdmajid@utem.edu.my

***Abstract*— Implementation of the Block Matching Algorithm (BMA) in Motion Estimation (ME) has been widely used in video encoder due to its simplicity and high compression efficiency. Many fast search methods of BMAs are being introduced to increase the efficiency of the ME process. This paper proposed a new algorithm, namely Adaptive Diamond Search Algorithm (ADS) which employs three different search patterns for its two main stages. At the initial step, an additional step is added to a predetermined static block to further speed up the search process as it is beneficial to small motion video sequence contents. The performances of the ADS are then compared with three selected established algorithms, namely the Full Search (FS), Diamond Search (DS) and Hexagon-Diamond Search (HDS). Based on the simulation result, the proposed algorithm yields a very good video quality performance with fewer search points compared with other algorithms.**

***Index Terms*— Adaptive Search, Diamond Search Pattern, Fast Block Matching Algorithm, Hexagon Search Pattern, Motion Estimation, Orthogonal Search Pattern**

I. Introduction

Among the popular and effective method in reducing the temporal redundancy is the Motion Estimation (ME) technique. ME is the process of analyzing successive frames in a video sequence to identify objects that are in motion. There are two main categories of the ME methods which are the feature-based method and the intensity-based methods. Moreover, the representation of the motion can be categorized into four different types, namely the global, pixel-based, block-based and region-based types [1]. The commonly used ME technique is the Block Matching Algorithm (BMA) used in most of the video codecs due to its simplicity and high compression efficiency [2]. BMA technique is basically a technique that finds the best motion vector (mv) on a block by block basis [3].

The latest international standard H.264/AVC has already adopted block motion estimation with full search algorithm into its codecs. However, due to its heavy computational complexity, the performances are greatly affected as the ME process consumed almost 80% of the total encoding time of the encoder [4].

A number of fast BMA has gradually increased in order to reduce the heavy computational cost while being able to maintain the accuracy degradation of ME. For example, Three-Step Search (TSS) [5], Four-Step Search (4SS) [6], Hexagon-Diamond Search (HDS) [7], Diamond Search (DS) [8], Cross Diamond Hexagonal Search (CDHS) [9], Cross Diamond Search (CDS) [10] and Orthogonal Search (OS) [11]. However, in real time video sequence, the algorithm may occasionally fall or trap into a local minimum, thus degrade the final video quality [11], [12].

Therefore, a new algorithm, namely the Adaptive Diamond Search (ADS) is proposed in order to improve the ME operation. The proposed algorithm together with 3 selected known algorithms are compared and their performances are analyzed in terms of peak signal-to-noise ratio (PSNR) and number of search points needed in order to determine their suitability to different motion content represented in respective video sequences.

II. Block Matching Algorithm

In BMA, the amount of motion is determined or estimated on a block by block basis. Typically, the current frame is firstly divided into macro-blocks of non-overlapping blocks with the size of M-by-N pixels. The block of pels in the current frame is compared to the corresponding block in the previous frame within a search area of size (M + 2q x N + 2q). The motion vector for each macro-block are estimated which is found by the best match based on certain matching cost function [13]. The general idea is shown in Figure 1, where p is the maximum displacement allowed, A is the block in the current frame, B is the block in the previous frame, and C is the search window in the previous frame [3].

Based on [13] the macro block is usually taken as M = 16 pixels and N = 16 pixels and the search parameter q = 7 pixels. A larger motion requires a larger q, and the larger the search parameter, the process of ME becomes more computationally expensive.

III. Adaptive Technique

For the manipulation of the adaptive techniques in Motion Estimation, there are several methods that can be used for this purpose. For example, paper [14] proposed an algorithm where it manipulates the search range to evaluate the influence on the bitrate and PSNR when the search range is set from a small range to a increasingly wide range. Meanwhile, paper [15] proposed an algorithm that implements an early termination technique that features an adaptive threshold based on statistical characteristics of rate-distortion (RD) cost regarding current block and previously processed blocks and modes. Therefore, for this paper, two techniques will be implemented into the proposed algorithm as mentioned below.

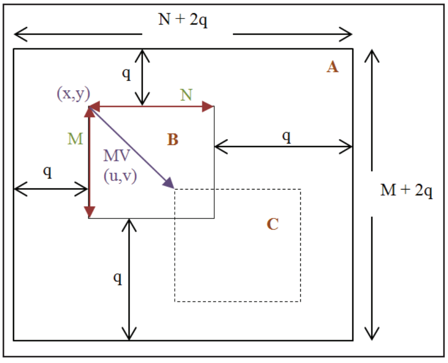

Figure 1: Current and previous frames (M x N) in a search window (N+2q x M+2q) and the motion vector (MV)

A. Inter-Mode Selection

The latest video code standard H.264/AVC supports variable block size modes which are divided into two types: 16x16, 16x8, 8x16 (macroblocks level) modes and 8x8, 8x4 and 4x8 (macroblock sub-level) modes. This option allows combinations of block sizes ranging from 16x16 pixels to 4x4 pixels within a macroblock for luminance component and correspondingly blocks of quarter sizes chrominance component [16] as shown in Figure 2.

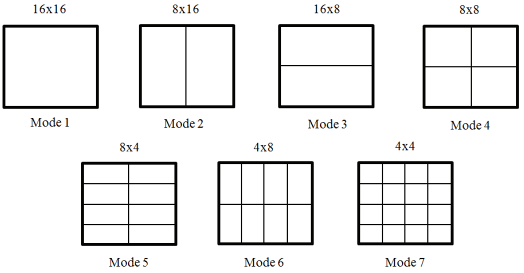

Figure 2: Variable block size of macroblock prediction mode

B. Early Termination Technique

Terminating the ME calculation at the initial stage of the process is one of the methods to reduce the computational complexity of ME process. Based on [17], a large percentage of zero-motion blocks occurred in many visual communication applications such as video telephony. This will cause a significant increase in the computational cost of the codecs. Therefore, paper [17] indicates that prejudgement can be made by computing matching errors between the current block and the reference block, and then comparing it with predetermined threshold, T. The current block then can be decided as a static block if the matching error is smaller than T, thus halting the search.

IV. EXISTING ALGORITHM

In the last few decades, several new fast BMAs have been developed and proposed for the purpose of reducing the computation of the FS algorithm in ME. Various techniques are proposed in the algorithms, such as the fixed search pattern, variable search range, hierarchical and multi resolution algorithm, sub-sampling, partial distortion elimination, spatio-temporal correlation and others [18]. They are the Diamond Search (DS) [19], Orthogonal Search (OS) [11], Hexagon-based Search (HEXBS) [20] and so on.

A. Diamond Search (DS) Algorithm

The DS algorithm implements a diamond shape into its technique as it tries to behave as an ideal circle-shaped coverage for mv investigation. The steps are summarized as follows [19]:

Step 1: A large diamond search pattern (LDSP) with nine search points centered at [0,0] of search window are tested. If the MCF point is at the center, proceed to Step 3. Otherwise, proceed to Step 2.

Step 2: The MCF point in Step 1 is re-positioned as the center of the new LDSP and all the points are tested. If the MCF point is at the center, proceed to Step 3. Otherwise, recursively repeat this step.

Step 3: The LDSP is then switched to a small diamond search pattern (SDSP) with 5 points. The MCF found in this step is the final mv which points to the best matching block.

B. Orthogonal Search Algorithm (OSA)

The orthogonal search algorithm (OSA) is first proposed by Puri et al. in 1987 [21] and it is a hybrid of the Three-Step Search and 2D Logarithmic Search. In this method, at each iteration with a logarithmic step search, four new locations are searched. At every step, two positions are searched alternately in the vertical and horizontal directions. It is stated that this algorithm is the fastest method of all known fast BMAs.

C. Hexagon-Based Search (HEXBS)

This algorithm implements the hexagonal pattern as its search pattern. The search procedure is as follows [20]:

Step 1: (Starting) The large hexagon with 7 points is centered at the origin [0,0] of the search window. If the MCF point is at the center, proceed to Step 3. Otherwise, proceed to Step 2.

Step 2: (Searching) A new large hexagon is formed with the previous MCF point at the center. Three new points are checked and if the MCF point is still at the center, proceed to Step 3. Otherwise, recursively repeat this step.

Step 3: (Ending) The large hexagon is then switched with a small hexagon with four points. All the points are evaluated to compare with the current MCF point. The new MCF point is the final solution of mv.

V. PROPOSED ALGORITHM

A new hybrid search pattern is proposed in order to reduce the computational complexity of the algorithm. The proposed algorithm employs three different search patterns, which are the Diamond Search (DS) together with the Orthogonal Search (OS) and the Hexagon Search (HS) to locate and refine the search process. The DS pattern is used to determine the best matching points, while the OS and HS patterns are used for refining the location of the final mv. The proposed algorithm is the extended and modified version of [22] for variable block size motion estimation. The flow chart of the operation is shown in Figure 3. The steps of the proposed algorithm are summarized as follows:

Step 1: All parameters involved are initialized first. The SAD between the current block and the block at the same location in the reference frame is computed and compared with a threshold, T=512. The value of threshold is chosen at T=512 because the average SAD of the static macroblocks (MBs) is within the range of 600 and 1300 [17]. It is to increase the search quality without causing any noticeable degradation on visual quality of the video. If the SAD_BLK is less than 512, the final MV is [0 0] and the search is stopped. Otherwise, proceed to step 2.

Step 2: Diamond search (DS) pattern is used in this step, as mentioned above. All the nine points of DS are tested to find the minimum cost function (MCF) point. All of the SAD calculated are compared to find the MIN_SAD_{DS}. Then, proceed to Step 3.

Step 3: If the MIN_SAD_{DS} is less than SAD_BLK, the algorithm is proceeded with the orthogonal search (OS). Mode 5 is used for the new search range and skip to Step 5. Otherwise, proceed to Step 4.

Step 4: If the MIN_SAD_{DS} is less than SAD_BLK + 512 but more than SAD_BLK, mode 4 is used for a new search range and proceeded with the hexagon search scheme. Otherwise, the MV= [0 0] and the search is stopped.

Step 5: The MV found is the final MV and the performance parameters are calculated and compared.

VI. RESULT AND ANALYSIS

The simulation of all the algorithms are performed and simulated using the MATLAB software for easy analysis and understanding. The test video sequences are chosen based on the type of motion contents as shown in the Table 1 below. The test sequences chosen are Akiyo, News and Foreman representing the Slow/Low motion, Moderate motion and Fast/Complex motion respectively [22, 25].

At the beginning of the stage, the block matching is conducted within the 15x15 search window and 16x16 block size, meanwhile the frame distance between predicted frame and original frame is set at 1. This is fixed for the calculation of the SAD_BLK at zero mv and for DS strategy. After that, the search window and block size are varied according to the level of the step search.

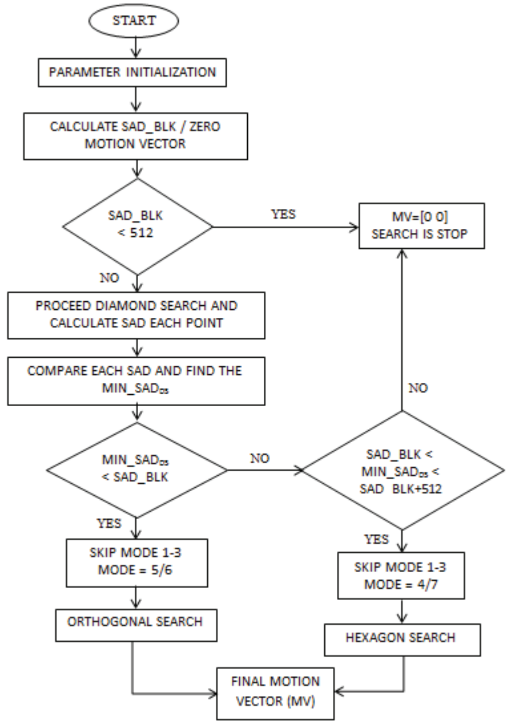

Figure 3: Flow chart of the proposed algorithm

Table 1
Test Sequence of Different Types of Motion Contents

Video sequences	Types of Motion	Format	Length	Frame Rate (fps)
Akiyo News	Slow	QCIF	100	15
Salesman Coastguard	Moderate			
Tennis Foreman	Fast/Complex			

The matching cost function (MCF) used in this paper is the Sum of Absolute Difference (SAD) [23] due to its simplicity and ease of implementation in real time compared to the Mean Squared Error (MSE) [23], Mean Absolute Difference (MAD) [24] and Mean Squared Difference (MSD) [24]. The equation is briefly explained as follows [24]:

$$SAD(i,j)=\sum_{k}\sum_{l}\left[C_f(k,l)-R_f(k+i,l+j)\right] \quad (1)$$

From the equation above, in the measurement, $C_f(k,l)$ is the location of the pel at the uppermost left in the block of the current frame, f while $R_f(k+i,l+j)$ is the location of the pel on the previous frame $f-1$, shifted by the (i,j) within the search area. The smallest $SAD(i, j)$ within the search area represents the best matching point.

Meanwhile, Peak Signal to Noise Ratio (PSNR) has been used as a benchmark and objective measure in evaluating the quality of the video metric. It is said in [23] that the higher the PSNR value, the better the quality of the compensated images. The formula of PSNR is defined as:

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

In the measurement, x_{ij} and $\hat{x}_{ij}$ are the pixel values of the reference and processed images respectively, and MN is the total number of pixels in the frame. The peak signal of the equation is 255 with 8-bit resolution while the noise is the mean squared difference (error) between the reference and current frames [18].

The performance criteria investigated in this project are the number of search points and the PSNR values. The performance of the proposed algorithm, Adaptive Diamond Search (ADS) and the selected algorithm, which are Full Search (FS), Diamond Search (DS) [7] and Hexagon-Diamond Seach (HDS) [8] are compared and tabulated in Table 2 and Table 3 below.

Table 2
Average Search Points Per Block Frame

Video	Algorithm			
	FS	DS	HDS	ADS
Akiyo	225	13.04	11.01	5.60
News	225	13.28	11.22	5.71
Foreman	225	16.59	12.57	6.14

Table 3
Average PSNR (dB) Per Block Frame

Video	Algorithm			
	FS	DS	HDS	ADS
Akiyo	43.25	42.39	42.28	43.48
News	35.27	34.26	34.26	36.43
Foreman	31.08	28.90	28.90	32.04

Table 2 shows the average number of search points needed to obtain the motion vector (MV) throughout the process. It can be seen from Table 2, that the ADS algorithm outperformed the other three algorithms with lesser search points for all of the three video sequences to obtain the final mv. These result shows the speedup ratio of the BMA required for the MV estimation.

As for Table 3, ADS algorithm achieved better PSNR value compared to the FS, DS and HDS algorithms. However, there is a slight degradation in quality for moderate and fast motion sequences compared to the small motion sequence.

Therefore, it can be said that the performance of average search points per block and PSNR values are ADS < HDS < DS < FS respectively.

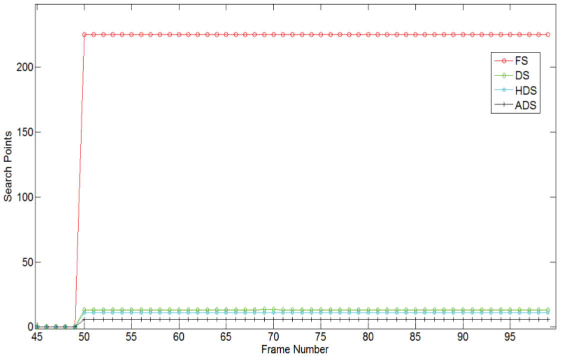

Figure 4: Comparative average search points per block per frame for "Akiyo" sequence (Class I)

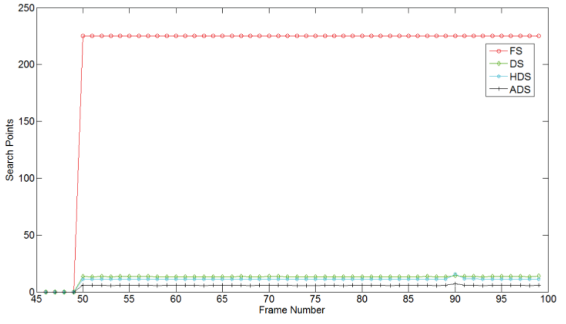

Figure 5: Comparative average search points per block per frame for "News" sequence (Class II)

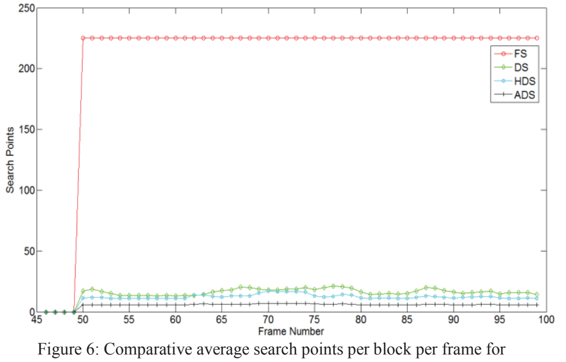

Figure 6: Comparative average search points per block per frame for "Foreman" sequence (Class II)

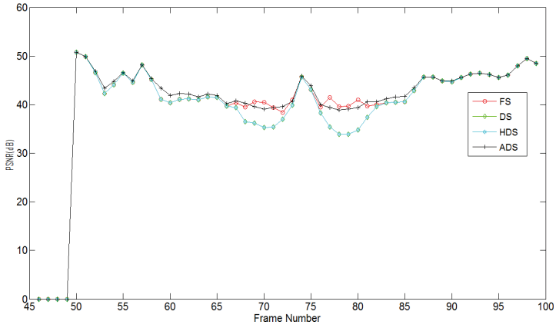

Figure 7: Comparative average PSNR per block per frame for "Akiyo" sequence (Class I)

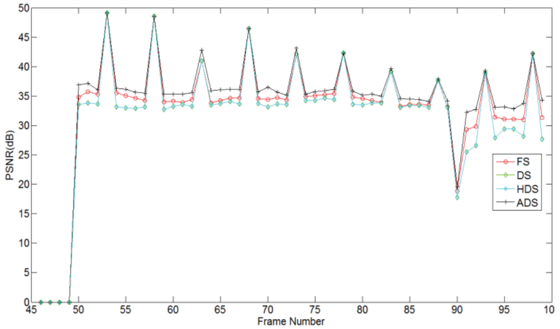

Figure 8: Comparative average PSNR per block per frame for "News" sequence (Class II)

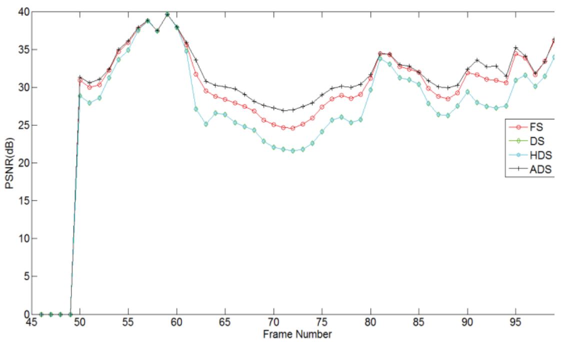

Figure 9: Comparative average PSNR per block per frame for "Foreman" sequence (Class III)

As for all of the above figures, it can be seen that based on Figure 4, Figure 5 and Figure 6, the proposed algorithm, namely the ADS gives the best result where most of the search points are below 10 point max for all motion types, and it is better than the others in terms of PSNR values based on Figure 7, Figure 8 and Figure 9.

VII. Conclusion

Adaptive Diamond Search (ADS) algorithm is proposed for a fast BMA motion estimation. Based on the experimental results, ADS performs better in terms of search points compared to other algorithms for all different types of motion contents. In terms of PSNR performance, ADS maintains a close performance with slightly higher PSNR value than FS, HDS and DS algorithms.

Acknowledgment

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] O. Marques, *Practical Image and Video Processing Using MATLAB*. Wiley, 2011.

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

[3] R. Rao and R. Srinivasan, "Predictive Coding Based on Efficient Motion Estimation," *IEEE Trans. Commun.*, vol. 33, no. 8, pp. 888–896, 1985.

[4] Z. Shi, W. A. C. Fernando, S. Member, and A. Kondoz, "Adaptive Direction Search Algorithms based on Motion Correlation for Block Motion Estimation," vol. 57, no. 3, pp. 1354–1361, 2011.

[5] N. Sun, C. Fan, and X. Xia, "An effective three-step search algorithm for motion estimation," *2009 IEEE Int. Symp. IT Med. Educ.*, pp. 400–403, Aug. 2009.

[6] L. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, 1996.

[7] R. A. Manap, S. S. S. Ranjit, A. A. Basari, and B. H. Ahmad, "Performance Analysis of Hexagon-Diamond Search Algorithm for Motion Estimation," *2010 2nd Int. Conf. Comput. Eng. Technol.*, pp. V3–155–V3–159, 2010.

[8] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation.," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–90, Jan. 2000.

[9] C. Cheung and L. Po, "Novel cross-diamond-hexagonal search algorithms for fast block motion estimation," *IEEE Trans. Multimed.*, vol. 7, no. 1, pp. 16–22, Feb. 2005.

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

[11] S. Soongsathitanon, W. L. Woo, and S. S. Dlay, "Fast search algorithms for video coding using orthogonal logarithmic search algorithm," *IEEE Trans. Consum. Electron.*, vol. 51, no. 2, pp. 552–559, May 2005.

[12] S. I. A. Pandian, B. A. George, and G. Josemin Bala, "A Study on Block Matching Algorithms for Motion Estimation," *Int. J. Comput. Sci. Eng.*, vol. 3, no. 1, pp. 34–44, 2011.

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

[14] T. S. T. Song, K. Ogata, K. Saito, and T. Shimamoto, "Adaptive Search Range Motion Estimation Algorithm for H.264/AVC," *2007 IEEE Int. Symp. Circuits Syst.*, pp. 3956–3959, 2007.

[15] M. G. Sarwer and Q. M. J. Wu, "Adaptive Variable Block-Size Early Motion Estimation Termination Algorithm for H.264/AVC Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, pp. 1196–1201, 2009.

[16] N. a. Khan, S. Masud, and a. Ahmad, "A variable block size motion estimation algorithm for real-time H.264 video encoding," *Signal Process. Image Commun.*, vol. 21, no. 4, pp. 306–315, Apr. 2006.

[17] Y. Nie and K.-K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation.," *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1442–9, Jan. 2002.

[18] M. Ghanbari, *Video coding : an introduction to standard codecs*, vol. 42. Institution of Electrical Engineers Stevenage, UK, UK, 1999.
[19] S. Zhu and K. K. Ma, “Correction to ‘a new diamond search algorithm for fast block-matching motion estimation’.,” *IEEE Trans. Image Process.*, vol. 9, no. 3, p. 525, Jan. 2000.
[20] C. Zhu, X. Lin, L. Chau, K. Lim, H. Ang, and O. Choo-yin, “A novel hexagon-based search algorithm for fast block motion estimation,” *2001 IEEE Int. Conf. Acoust. Speech, Signal Process. Proc. (Cat. No.01CH37221)*, vol. 3, pp. 1593–1596.
[21] A. Puri, H. Hang, D. L. Schilling, and T. B. Laboratories, “An Efficient Block-Matching Algorithm for Motion-Compensated Coding,” *IEEE Int. Conf. Acoust. Speech, Signal Process.*, 1987.
[22] N. A. Hamid, A. M. Darsono, N. A. Manap, R. A. Manap, and H. A. Sulaiman, “A New Orthogonal – Diamond Search Algorithm for Motion Estimation,” in *International Conference on Computer, Communications, and Control Technology (I4CT) 2014*, 2014, no. I4ct, pp. 467–471.
[23] S. Metkar and S. Talbar, *Motion Estimation Techniques for Digital Video Coding*. Springer India, 2013.
[24] S. Usama, M. Montaser, and O. Ahmed, “A Complexity and Quality Evaluation of Block Based Motion Estimation Algorithms,” *Acta Polytech.*, vol. 45, no. 1, 2005.
[25] N. A. Hamid , A. M. Darsono, N. A. Manap, R. A. Manap " Performance Evaluation Of Orthogonal-Diamond Search Of Block Matching Algorithm For Video Coding" *Jurnal Teknologi*, vol. 76, no. 1, pp. 61-66, 2015.