Jurnal
Teknologi

Full Paper

PERFORMANCE EVALUATION OF ORTHOGONAL-DIAMOND SEARCH OF BLOCK MATCHING ALGORITHM FOR VIDEO CODING

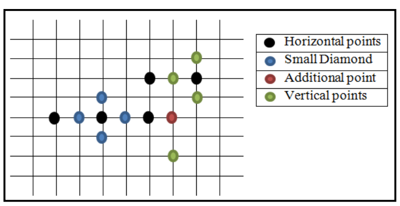
Nurul 'Atiqah Hamid*, Abd Majid Darsono, Nurulfajar Abd Manap, Redzuan Abd Manap

Center for Telecommunication Research and Innovation (CeTRI), FKEKK, University Teknikal Malaysia Melaka, Melaka, Malaysia

Article history
Received
5 December 2014
Received in revised form
18 March 2015
Accepted
1 August 2015

*Corresponding author
m021310019@student.utem.edu.my

Graphical abstract



Abstract

Several drawbacks of established fast Block Matching Algorithm (BMA) are the reasons why new fast BMAs are being developed and proposed in these recent years in order to reduce the computational cost while maintaining the quality of the video signal. In this paper, a new algorithm is proposed, namely Orthogonal-Diamond Search (ODS) which employs an orthogonal-shaped search pattern in the first step and then is switched into diamond-shaped search pattern for the next step. Few established algorithm, namely Orthogonal Search (OS), Full Search (FS), Diamond Search (DS) and Hexagon-Diamond Search (HDS) are implemented using MATLAB along with the ODS and their performance are being compared and analyzed in terms of computational complexity, peak signal-to-noise ratio (PSNR), and number of search points. Simulation result shows that the proposed algorithm can find motion vector with fewer number of search points while maintains close performance of video quality with other selected algorithms.

Keywords: Fast block matching algorithm, motion estimation, orthogonal diamond search algorithm

Abstrak

Beberapa kelemahan yang mantap cepat blok sepadan algoritma (BMA) adalah antara sebab mengapa BMAs cepat baru sedang dibangunkan dan dicadangkan dalam beberapa tahun kebelakangan ini untuk mengurangkan kos pengiraan sambil mengekalkan kualiti isyarat video. Dalam kertas ini, algoritma yang baru dicadangkan, iaitu Orthogonal-Diamond Cari (ODS) yang menggunakan satu pola carian orthogonal berbentuk dalam langkah pertama dan kemudian dihidupkan ke dalam pola carian berbentuk diamond untuk langkah seterusnya. Sedikit ditubuhkan algoritma, iaitu Cari Orthogonal (OS), Cari penuh (FS), Berlian Cari (DS) dan Heksagon-Berlian Cari (HDS) dilaksanakan menggunakan MATLAB bersama-sama dengan ODS yang dan prestasi mereka sedang berbanding dan dianalisis segi kekompleksan pengiraan, Nisbah isyarat-ke-hingar puncak (PSNR) dan bilangan Mata Cari. Hasil simulasi menunjukkan bahawa algoritma yang dicadangkan boleh menemui vektor gerakan dengan bilangan Cari mata sementara mengekalkan prestasi rapat kualiti video oleh algoritma terpilih lain.

Kata kunci: Block cepat sepadan algoritma, anggaran gerakan, algoritma carian ortogon berlian

1.0 INTRODUCTION

The demand for fast and effective method for video compression is rapidly increasing. Motion Estimation (ME) is considered as one of the popular and effective method in reducing the temporal redundancy between successive frames of a video sequence. ME estimate the motion by finding the motion vectors of the objects in an image sequence [1]. Block matching based ME or known as Block Matching Algorithm (BMA), the most commonly used for ME method and widely used in most of the video codecs, including the H.26x series [2], [3] due to its implementation simplicity and high compression efficiency [4].

The simplest BMA is the Full Search (FS) algorithm which exhaustively checking all the possible displacement within the search window to find the best matching block [1]. However, this leads to high computational complexity, thus making it not a good choice for real-time video coding implementation [5].

Therefore, this leads to the developments of various fast BMAs 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) [5], Four-Step Search (4SS) [6], New-Three-Step Search (NTSS) [7], Diamond Search (DS) [8], Hexagon-Diamond Search (HDS) [9], Cross Diamond Search (CDS) [10], Cross Diamond Hexagonal Search (CDHS) [11] and Orthogonal Search (OS) [12].

In the real-time video sequences, the distribution of motion vector (MV) is highly center-biased. This leads to development of center biased BMAs which eventually provides the close prediction accuracy, especially for slow motion video sequences. TSS and OS are more efficient in finding the global minimum, especially for those sequences with large motion. However, it is becoming inefficient if for small motion as it tends to be trapped into local minimum [12], [13].

In this paper, Orthogonal-Diamond Search (ODS) is proposed and implemented onto standard video sequences using MATLAB with few established BMA algorithm namely FS, TSS, DS and HDS. 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.

BMA is best described as the technique that estimates the amount of motion on a block by block basis. Block matching finds the best block from the previous frame to reconstruct an area of the current frame by dividing each coding frame into non-overlapping blocks with size of N-by-M pixels. The macro blocks are then compared with corresponding block and its adjacent neighbors in the previous frame within a search area of size $(N + 2q \times M + 2q)$. 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 [14].

Based on [15], usually the macro block is taken as N = 16 pixels and M = 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.

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 and previous block which is found by the best match based on certain matching cost function [15].

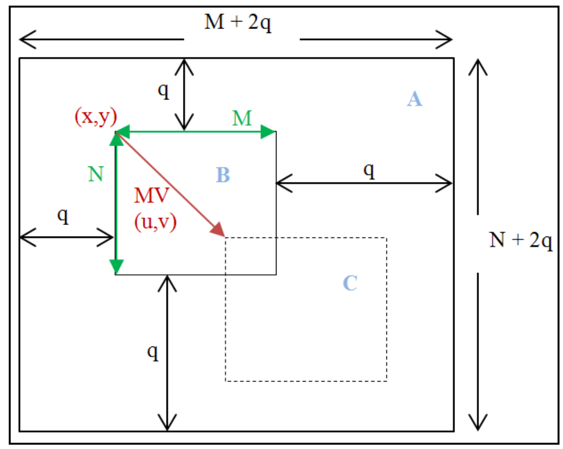


Figure 1 Current and previous frames (N x M) in a search window (M+2q x N+2q) and the motion vector (MV)

2.0 MATERIAL AND METHOD

2.1 Matching Cost Function

There are several matching cost functions that can be used for this purpose, such as mean squared error (MSE) [16], sum of absolute difference (SAD) [16], mean squared difference (MSD) [17] and mean absolute different (MAD) [17]. One of the major factors affecting the ME algorithm complexity and its performance is the cost function. MSD and MAD 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 [17]:

Mean Squared Difference (MSD)

$$MSD(i,j)= \frac{1}{NM}\sum_{k}\sum_{l}[C_f(k,l)-R_f(k+i,l+j)]^2$$

Mean Absolute Difference (MAD)

$$MAD(i,j)= \frac{1}{NM}\sum_{k}\sum_{l} C_f(k,l)-R_f(k+i,l+j)$$

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 $R_f(k+i,l+j)$ is the location of the pel on the previous frame f-1, shifted by the (i,j) within the search area.

Meanwhile, NM is the block size and the smallest $\mathrm{MSD}(\mathrm{i},\mathrm{j})$ or $\mathrm{MAD}(\mathrm{i},\mathrm{j})$ within the search area represents the best match.

2.2 Peak Signal-to-Noise Ratio

The Peak Signal-to-Noise ratio (PSNR) is used to determine the quality of the compressed images. The PSNR equation for a grayscale image is defined as follows [18]:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{(\text{Peak to peak value of original data})^2}{\mathrm{MSD}}\right)$$

$$= 10\log_{10}\left(\frac{(255)^2}{\frac{1}{\mathrm{NxM}}\sum_{i=1}^{N}\sum_{j=1}^{M}\left(x_{ij} - \hat{x}_{ij}\right)^2}\right)$$

Where (NM) 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). The higher the PSNR value, the better the quality of the compensated images [17].

2.3 Orthogonal Diamond Search Algorithm

Orthogonal Diamond Search algorithm (ODS) employs the orthogonal shape and Small Diamond Search Pattern (SDSP) for its search steps. Although the orthogonal search algorithm (OSA) has only 13 search points per block ME, it uses uniformly allocated search patterns in its first few steps which are inefficient for catching small MV of stationary or quasi-stationary blocks. Therefore center-biased search point pattern is used to improve compensation error [19].

The frame distance between predicted frame and original frame is set to be 1 for consistent comparison with previous research works with 16x16 block size. The block matching is conducted within the 15x15 search window size. The maximum displacement is set with to be ±7 horizontally and vertically. Instead of MSD, the matching cost function used is MAD in the procedure as it does not require a multiplication operation, hence reducing the block matching computational requirement [10].

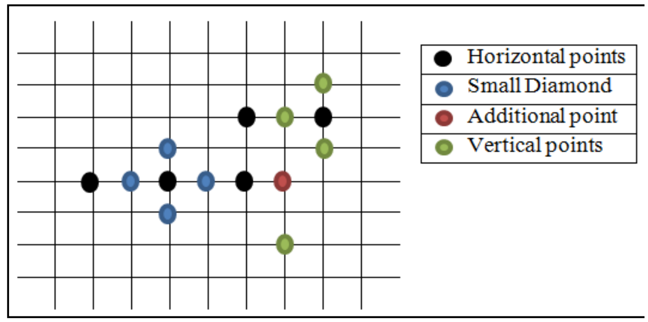


Figure 2 The proposed Orthogonal Diamond search pattern

The proposed ODS algorithm steps are summarized as follows:

Step 1: The first three points initiate the orthogonal search pattern where the center point is centered at the origin of the search window 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 2. Otherwise, the small diamond search pattern (SDSP) is employed to the center point and one additional point is added and proceeds to Step 2.

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

Step 3: The step size is reduced to 1 point for the two horizontal search points and then proceeds to Step 4.

Step 4: The search pattern and step size is the same as Step 3 but in vertical direction and the MV is found in this step is the final MV.

3.0 RESULTS AND DISCUSSION

As for the simulation, only the first 40 frames of total frames of the available test sequences are considered to be simulated and analyzed in this paper to serve an easier understanding of the analysis done. The test sequences used for this experiment are Akiyo, News and Foreman video sequences which consists of slow motion, moderate motion and fast motion respectively [17].

The performance is compared between three established algorithms, namely Full Search (FS), Diamond Search (DS), Hexagon-Diamond Search (HDS) with the proposed algorithms Orthogonal Diamond Search (ODS) algorithm. 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	Algorithms			
	FS	DS	HDS	ODS
Akiyo	225	13.02	11.00	10.00
News	225	13.07	11.03	10.02
Foreman	225	16.75	11.77	10.45

Table 2 Average PSNR per Block Frame

Video	Algorithms			
	FS	DS	HDS	ODS
Akiyo	43.30	42.90	42.90	42.90
News	37.82	37.28	37.28	37.28
Foreman	29.76	27.92	27.92	27.92

Table 1 shows the average number of search pattern required in obtaining MV per block per frame. The average search point shows the speedup ratio of the BMA required for the MV estimation. Based on the table, ODS algorithm contributes lesser search points compared to other three algorithms. The FS algorithm gives the highest average search points as it searches all possible search points within the search window to finding the optimum minimum point. Therefore, it can be said that the average search points per block are ODS < HDS < DS < FS respectively.

As for Table 2, the average PSNR per block per frame are tabulated for involved algorithms. It can be seen that DS, HDS and ODS gives similar PSNR value while FS gives the highest PSNR value. There is a slight degradation in quality for moderate and fast motion sequences compared to the small motion sequence.

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.

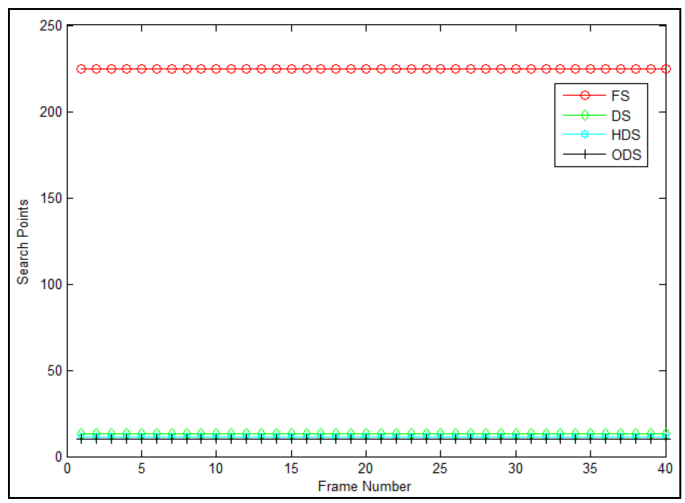


Figure 3 Comparative average search points per block per frame for "Akiyo" sequence

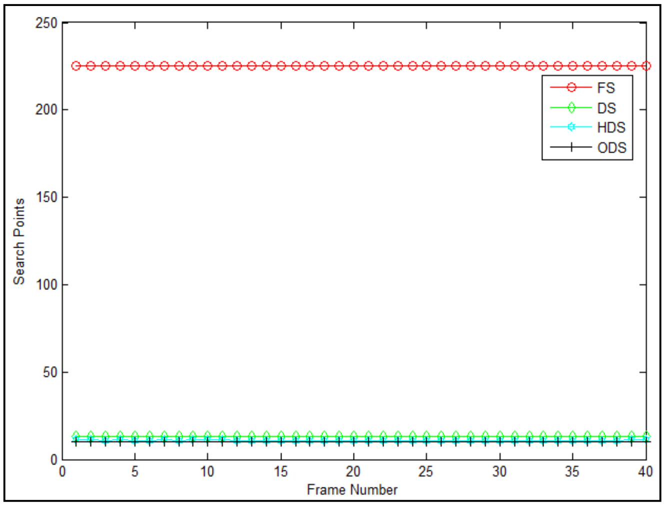


Figure 4 Comparative average search points per block per frame for "News" sequence

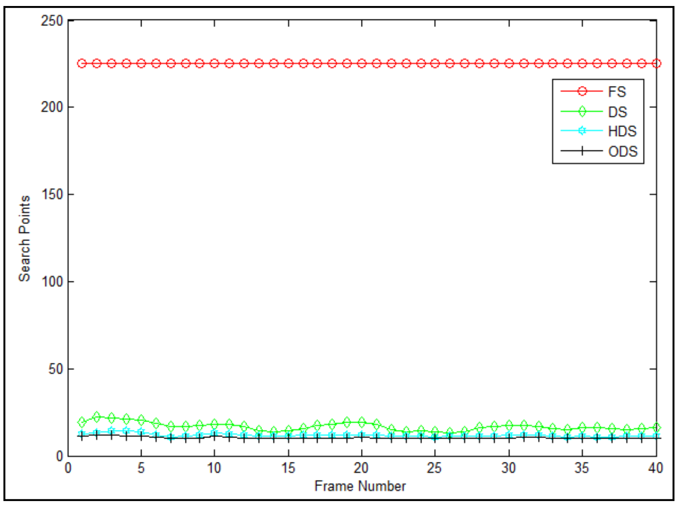


Figure 5 Comparative average search points per block per frame for "Foreman" sequence

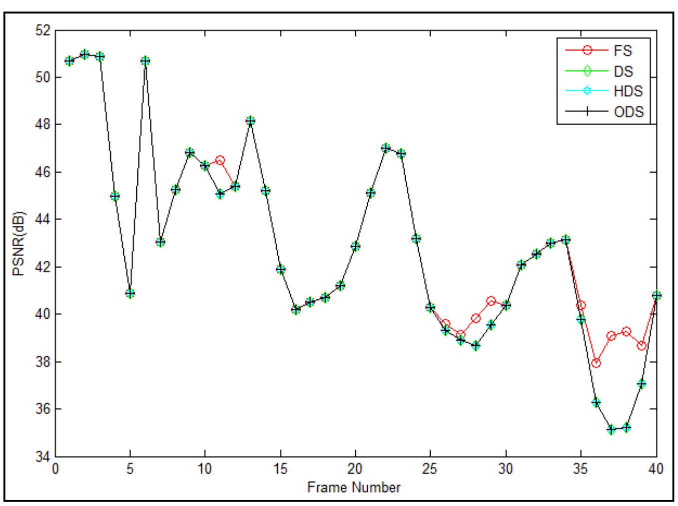


Figure 6 Comparative average PSNR per block per frame for "Akiyo" sequence

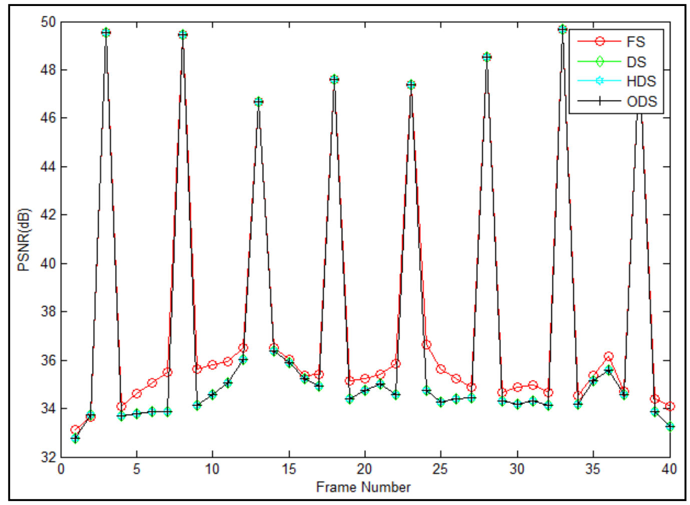


Figure 7 Comparative average PSNR per block per frame for "News" sequence

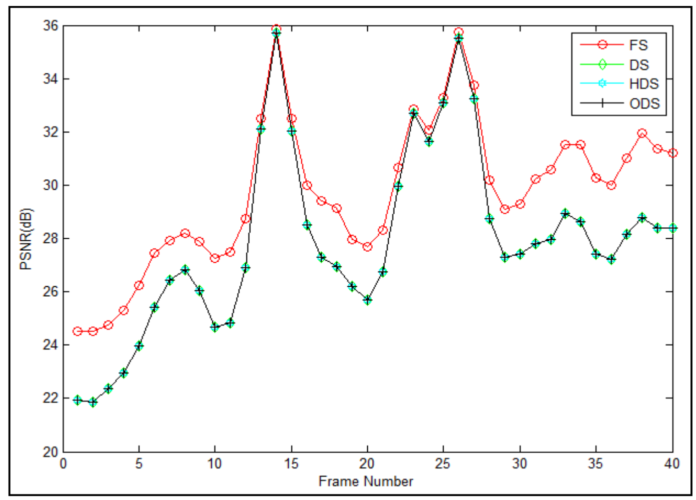


Figure 8 Comparative average PSNR per block per frame for "Foreman" sequence

From the figures above, Figure 3, Figure 4 and Figure 5 show that ODS algorithm gives the best result in terms of search points over other algorithms where most points for each frame are below 10 point max for all types of motions. However, as shown in Figure 6, Figure 7 and Figure 8, the ODS algorithm and other algorithms give similar and close performance towards each other in terms of PSNR performances.

4.0 CONCLUSION

Orthogonal Diamond Search (ODS) algorithm is being proposed for the fast BMA motion estimation. Based on the experimental results, ODS 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, ODS 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 (ERGS/1/2013/TK02/UTEM/03/04/E00020) Exploratory Research Grant Scheme that makes this research work possible.

References

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

[2] Rijkse, K. 1996. H.263: Video Coding for Low-Bit-Rate Communication. *IEEE Communications Magazine*. 34(12): 42-45.

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

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

[5] Sun, N., Fan, C. & Xia, X. 2009. An Effective Three-Step Search Algorithm for Motion Estimation. *2009 IEEE International Symposium on IT in Medicine & Education*. 400-403.

[6] Po, L. & 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): 313–317.

[7] 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).

[8] Zhu, S. & Ma, K. K. 2000. Correction to "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation". *IEEE Transactions On Image Processing: A Publication of the IEEE Signal Processing Society*. 9(3): 525.

[9] 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*. V3–155–V3–159.

[10] Jia, H. & Li, Z. 2004. A New Cross Diamond Search Algorithm For Block Motion Estimation. In *IEEE International Conference on Acoustics Speech and Signal Processing*. Iii: 357-60.

[11] Cheung, C. & Po, L. 2005. Novel Cross-Diamond-hexagonal Search Algorithms for Fast Block Motion Estimation. *IEEE Transactions on Multimedia*. 7(1): 16-22.

[12] Soongsathitanon, S., Woo, W. L. & Dlay, S. S. 2005. Fast Search Algorithms for Video Coding Using Orthogonal Logarithmic Search Algorithm. *IEEE Transactions on Consumer Electronics*. 51(2): 552-559.

[13] Immanuel Alex Pandian, S., Josemin Bala, G. & Anitha, J. 2011. Enhanced Modified Orthogonal Search for Motion Estimation. *2011 IEEE Recent Advances in Intelligent Computational Systems*. 907-910.

[14] Rao, R. & Srinivasan, R. 1985. Predictive Coding Based on Efficient Motion Estimation. *IEEE Transaction on Communications*. 33(8): 888-896.

[15] 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.

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

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

[18] Po, L.-M. & Cheung, C. 1996. A New Center-biased Orthogonal Search Algorithm for Fast Block Motion Estimation. In *IEEE TENCON. Digital Signal Processing Applications*. Perth: IEEE. 874-877.

[19] Metkar, S. & Talbar, S. 2013. *Motion Estimation Techniques for Digital Video Coding*, Springer India.