International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.4 (2014), pp.317-324
http://dx.doi.org/10.14257/ijsip.2014.7.4.30

Block based Motion Estimation using Octagon and Square Pattern

S. Sowmyayani[1] and P. Arockia Jansi Rani[2]
[1]*Research Scholar, Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India*
[1]*sowmyayani@gmail.com*
[2]*Assistant Professor, Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India*
[2]*jansi_msu@yahoo.co.in*

Abstract

This paper proposes a motion estimation algorithm that uses different search pattern. Motion Estimation Algorithm plays a vital role in video compression. In block-based motion estimation, search pattern with different shape or size has very important impact on search speed and distortion performance. In this paper, Octagon and square search pattern is proposed for block-based motion estimation. The proposed method contains 13 points that structures as Octagon for finding large motion vectors. In addition, Square Search Pattern with 8 points is used for finding small motion vectors. The speedup gain of the proposed method over the Unsymmetrical-cross Multi-hexagon-grid Search (UMHexS) algorithm is more striking. Experimental results substantially proved that the performance of the proposed method is better than the existing state-of-art motion estimation methods.

Keywords: *motion estimation, search patterns, full search method, Block Distortion Measure.*

1. Introduction

There are two types of motion estimation methods namely pixel-based motion estimation and Block-based motion estimation. The pixel-based motion estimation approach seeks to determine motion vectors (MV) for every pixel in the image. This is also referred to as the ‘optical flow method’, which works on the fundamental assumption of brightness constancy, that is, the intensity of a pixel remains constant when it is displaced. However, no unique match for a pixel in the reference frame is found in the direction normal to the intensity gradient. It is for this reason that an additional constraint is also introduced in terms of the smoothness of velocity (or displacement) vectors in the neighborhood. The smoothness constraint makes the algorithm interactive and requires excessively large computation time, making it unsuitable for practical and real-time implementation. An alternative and faster approach is the block-based motion estimation. In this method, the candidate frame is divided into non-overlapping blocks (of size16 x 16, or 8 x 8, or even 4 x 4 pixels in the recent standards) and for each such candidate block, the best motion vector is determined in the reference frame. [17]

The block matching algorithms that have been implemented earlier are Full Search(FS)[1], Three Step Search(TSS) [18], New Three Step Search(NTSS) [2], Simple and Efficient TSS(SES) [3], Four Step Search(4SS) [4], Cross Search Pattern [5], Diamond Search(DS)[6], Cross Diamond Search(CDS)[7], Hexagon-based Search(HEXS) [8], hexagon-diamond search (HDS) [9], Octagon-based Search(OCTBS) [11], UMHexagonS [12] *etc.*

Full search is the initial block search techniques where for getting the best match in the reference frame it checks the current block with whole candidate block. This algorithm

ISSN: 2005-4254 IJSIP

takes more time so that it is not feasible to large sequence video [19]. Koga et al. [18] proposed TSS which employs consistently allocated checking point pattern in the 1st step which is not realistic for small motion estimation. In 1993, NTSS algorithm is proposed which is having center biased characteristics which is efficiently used for small motion vector [2]. In NTSS half way stop techniques are employed for stationary & quasi-stationary block. This NTSS is more robust than the TSS which gives the more accurate smaller motion estimation. In 1996, 4SS method is proposed by Lai-man Po et al. [6], this algorithm also works on the center-biased technique and half way search techniques is used. This algorithm also provides the better result than the TSS and gives similar performance with the NTSS in terms of Mean square error with minimum computational cost. In 2000 New diamond search algorithm is proposed which gives better result than the TSS & 4SS [6]. In 2001 Ce Zhu et al. [8] proposed a novel hexagonal based search algorithm (HEXBS). This algorithm gives the motion vector with minimum search points than the DS. It also gives closer performance with the NTSS. In 2002 Chun –Ho Cheung et al. [9] proposed Cross Diamond Search Algorithm which includes cross search pattern as the first step and then large & small search sample. This CDS gives 40% enhancement on speedup gain.

In 2003, Efficient three step search (ETSS) is proposed by Lap-Pui Chau et al. [10] small diamond checking point pattern & the unrestricted search step is employed to search the center biased motion vectors which gives the better performance than the TSS, N3SS, 4SS in terms of MSE with lesser search points. Octagon-based Search is proposed by L.P.Chau and C.Zhu [11]. This algorithm uses 9 Search points including center point. The advantage of using this search pattern is that it can search a relatively large area with very few search locations and thus this pattern has a good efficiency. The UMHexagonS is one of the best fast Motion Estimation algorithm in H.264/AVC (Advanced Video Coding) reference software. It achieves an excellent coding performance by using multiple initial search point predictors and hybrid block matching search pattern. This pattern makes the computational complexity of Motion Estimation (ME) increased [12]. In 2004 Enhanced Hexagonal search algorithm is proposed by Ce Zhu et al. [14]. This algorithm provides the improvement in HEXBS in the form of reducing the number of search points and distortion. In 2009 Cross hexagon based algorithm is proposed by Li Hong ye. In this algorithm cross pattern & hexagonal algorithm is combined & it gives the better result than the hexagonal search [15]. In 2010 R.A. Manap *et al.*, proposed hexagon-diamond search (HDS). The HDS algorithm gives the better result than the NTSS, FS [19]. In 2012, a fast Octagon and triangle search pattern is proposed by C. Duanmu and Yu Zhang. The multi-octagon-grid search pattern (OCTTS) is used in this algorithm. This pattern has much less computational complexity than that of the multi-hexagon-grid search pattern in the UMHexagonS algorithm and this pattern has good search efficiency [13]. Motion estimation algorithm is useful in video compression [16].

The proposed work includes Octagon search pattern for large motion and square search pattern for small motion. This method gives better performance than other motion estimation algorithms.

2. System Architecture

The overall system consists two subsystems: Motion Estimation and Motion Compensation. The two subsystems are shown in Figure 1 and Figure 2. In Motion Estimation, video sequence is divided into frames. Each current frame is again subdivided into macroblocks. For each current macroblock, motion vector is estimated using Octagon and Square Search pattern. Motion Estimation uses reference frame. In Motion Compensation, each macroblock in the current frame is reconstructed with its reference frame using motion vectors. The PSNR is calculated with original frame and its reconstructed frame.

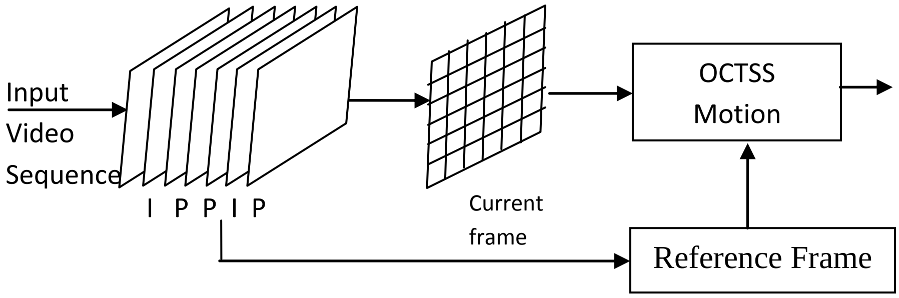


Figure 1. Motion Estimation

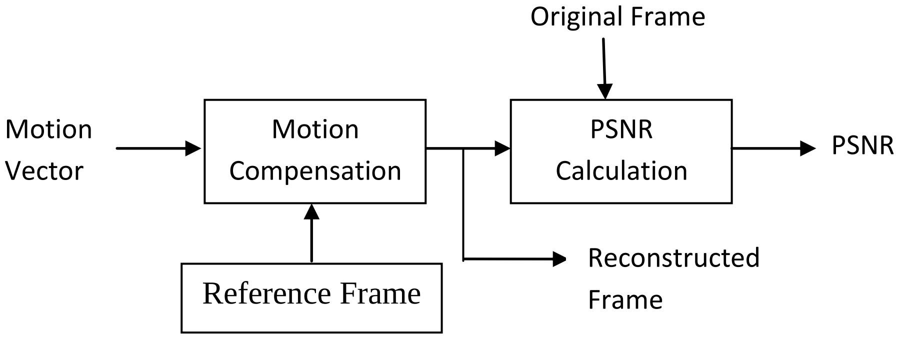


Figure 2. Motion Compensation

3. Octagon and Square Search Pattern

Each block in the current frame is searched in a particular range which is called search window in the reference frame. The maximum displacement in the search window is ±7 pixels in both the horizontal and the vertical directions for 16 x 16 macroblock size. Full search method searches all the points in the search window. Since FS uses more search points, it achieves best match for the macroblock in the current frame. But it has high computational cost. As search points increases, best match will be found. DS uses nine search points, while HS uses 7 points. CDS requires about two to five points fewer than DS. OCTSS uses 13 search points for large motion and 8 search points for small motion. The search points are shown in Figure 3.

In the searching procedure of the OCTSS algorithm, initially, Block Distance Measure (BDM) is calculated for center point. If it is zero, there is no motion in the frame. So, set motion vector to (0,0) and go to next macroblock. Otherwise, calculate BDM for all points (13 for octagon and 8 for square pattern) in OCTSS pattern. The search pattern is then switched from Octagon pattern to Square pattern. Square search is repeated until the minimum BDM occurs at the center point.

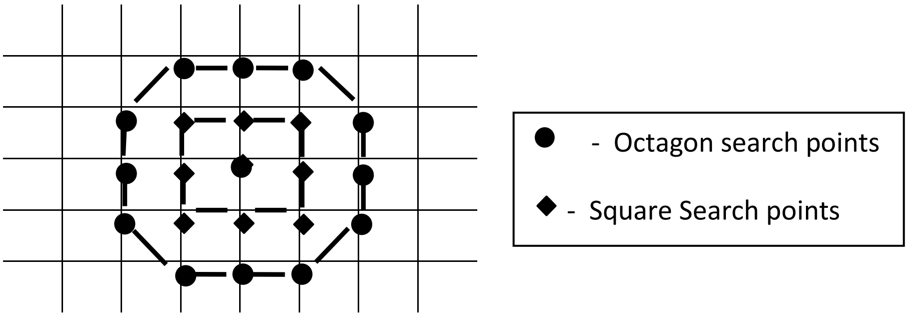


Figure 3. Octagon and Square Search Pattern

Step by step process is explained below.

Step: 1 Calculate BDM for center point. If it is zero, go to Step 4

Step: 2 Calculate BDM for all points (13 for octagon+8 for square). Move the center to this point which has minimum BDM.

Step: 3 The search pattern strategy is square from this step. Calculate BDM and move to this point with minimum BDM. Repeat Step 3 till center point has minimum BDM.

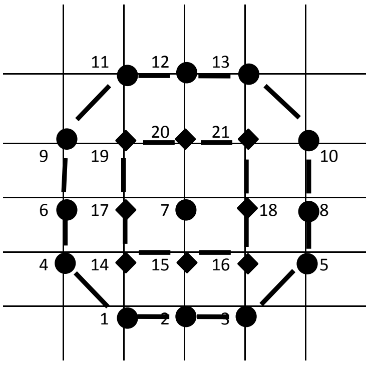
Step: 4 Stop the search.

Matching of one macro block with other is based on the output of the BDM. The candidate macro block that results in the least BDM is the one that matches closest to the current block. Among the various cost functions, the one that is less computationally expensive is the Mean Absolute difference (MAD) [15] and is given by the formula:

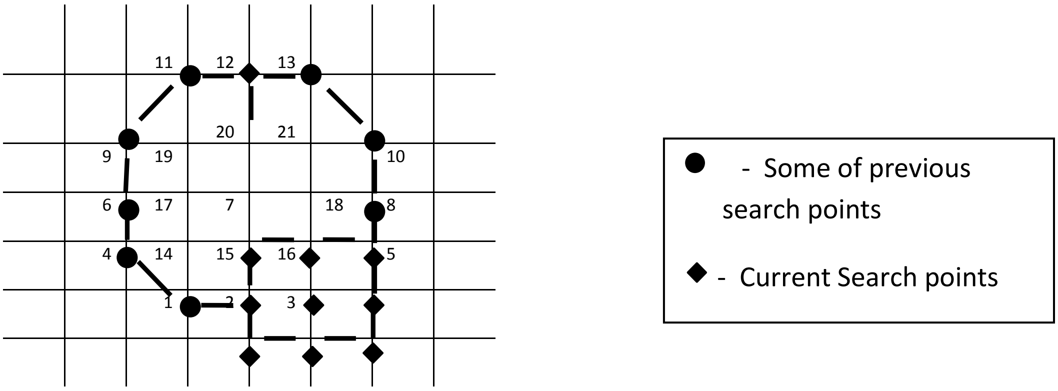
$$MAD = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|C_{ij}-R_{ij}\right| \quad (1)$$

where M and N is the size of the macro block, C_{ij} and R_{ij} are the pixels being compared in current macro block and reference macro block, respectively.

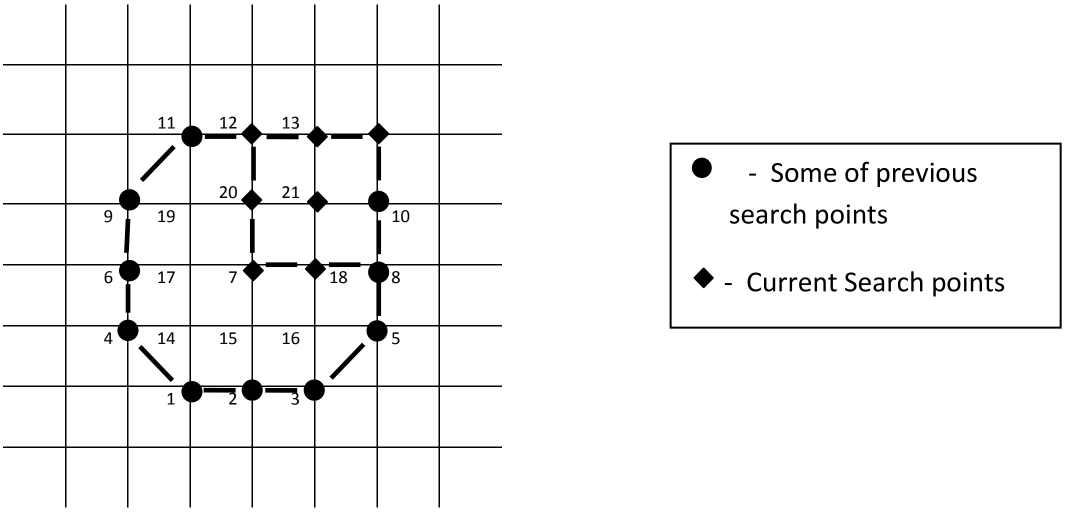
The process of Octagon search is illustrated in Figure 4. Search points are numbered for clear illustration. In Figure 4(a) search points are numbered, octagon search pattern is number from 1 to 13. Square search pattern is number from 14 to 21. All search points are used for pattern matching assuming the BDM of the center point is not zero. If the BDM of the center point is zero, searching process is stopped. If the search point numbered 3 has minimum BDM, next search pattern is shown in Figure 4(b). If the search point numbered 21 has minimum BDM, next search pattern is shown in Figure 4(c).



(a)



(b)



(c)

Figure 4. (a) Search points are numbered, (b) Searching pattern if point 3 has minimum BDM in (a), (c)Searching pattern if point 21 has minimum BDM in (a)

4. Experimental Results

Experimental results show how OCTSS is used for large motion video sequence and small motion video sequence. OCTSS are tested with all the three test video sequences. The “Foreman “series is the fastest as it has less movement followed by “Soccer” series. The “Mobile “sequence is relatively slow due to the nature of the video with high relative motions and movements in the background. In the experiments, the block size is fixed at 16x16. To make a consistent comparison, block matching is conducted within a15x15 search window (*i.e.*, 7 pels displacement in horizontal and vertical directions).

The quality of the reconstructed sequence should be estimated by subjective tests. One of the subjective metrics is Mean Square Error (MSE) which is evaluated between original frame and reconstructed frame. The lesser the MSE value, the better the prediction quality.

Mean Square Error is given by

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(f(M,N) - f'(M,N))^2 \qquad (2)$$

where f(M, N) represents the current frame and f ‘(M, N) is the reconstructed frame with frame size as M x N. Another widely used metric for comparing various image compression techniques is the peak-signal-to-noise-ratio (PSNR). The mathematical formulae for PSNR is

$$PSNR = 10\,log_{10}\left[\frac{(2b-1)^2}{MSE}\right] \qquad (3)$$

The b in the equation is the number of bits in a pixel. For 8-bit uniformly quantized video sequence b = 8. The higher the PSNR value, the better the quality of the compensated image.

Results for mobile sequence (QCIF – 300 frames) are shown in Table I. Experiments were also performed for Foreman sequence (QCIF – 300 frames) and soccer sequence (QCIF – 150 frames). All sequences are compared with existing best known Motion Estimation Algorithms.

In addition to Average PSNR per frame and Average MSE/pixel, Computational time (Speed), Number of search points per macroblock (Ns), Distance and Probability are calculated. The computational time has been calculated on Intel core-2-duo processor (3 GHz). Distance is calculated as the average distance from MV of the proposed work to the true MV. The probability of achieving the true MV, which is obtained using FS is also calculated.

From Table 1, it is found that the full search (FS) technique produces better quality image as it gives better performance in PSNR calculation, but takes larger number of search points.

Table 1. Results for Foreman Sequence (QCIF – 300 frames)

Methods	Speed (ms)	Ns	Distance	Probability	Average MSE/pixel	Average PSNR(dB)
FS	0.0285	195.3076	0	1	0.6854	27.6510
TSS	0.0980	22.4254	0.0207	0.9923	0.6858	27.6245
NTSS	0.0721	16.1855	0.0146	0.9938	0.6856	27.6456
SESTSS	0.0825	16.6052	0.0280	0.9879	0.6884	27.6150
4SS	0.0697	15.5731	0.0171	0.9966	0.6854	27.6355
DS	0.0847	12.3978	0.0169	0.9935	0.6854	27.6465
ARPS	0.0578	5.8891	0.0183	0.9929	0.6857	27.6433
HEXS	0.0774	10.6421	0.0285	0.9836	0.6927	27.5980
UMHexagonS	0.2654	9.3695	0.0254	0.9899	0.6900	27.6401
OCTS	0.0825	11.8074	0.0919	0.9227	0.6854	27.5096
OCTTS	0.0238	10.8965	0.0246	0.9914	0.6854	27.6235
Proposed OCTSS	0.0215	9.0563	0.0175	0.9930	0.6854	27.6511

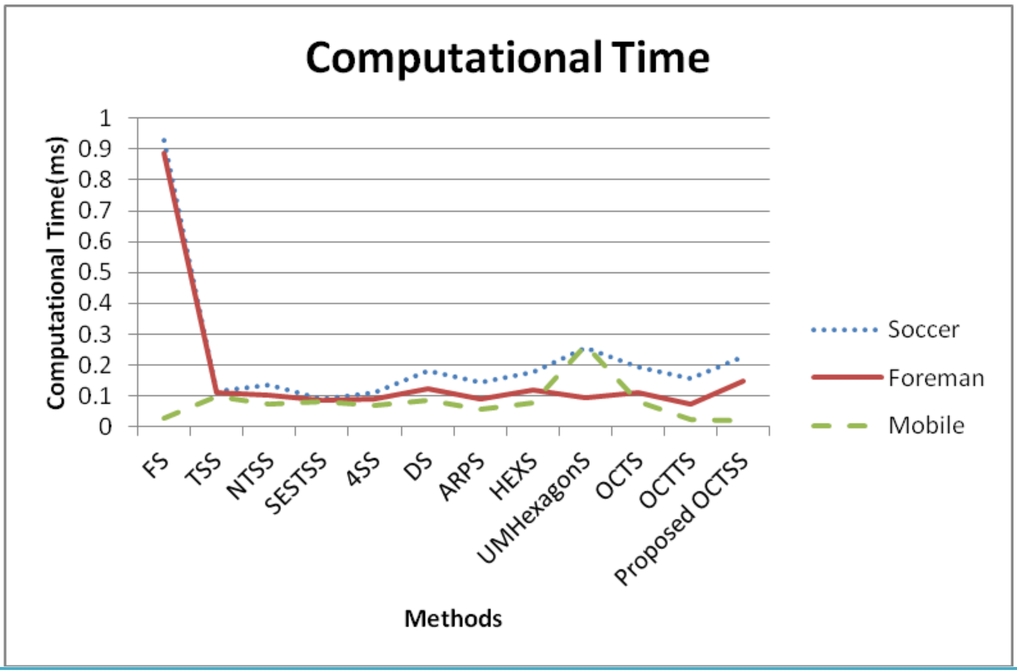


Figure 5. Computational Time of various methods for Mobile, Foreman and Soccer Sequences

From Figure 5, computational time of Full Search method is higher than all other methods in case of faster sequences. Hence it is enough to compare proposed work with all other methods except Full Search Method.

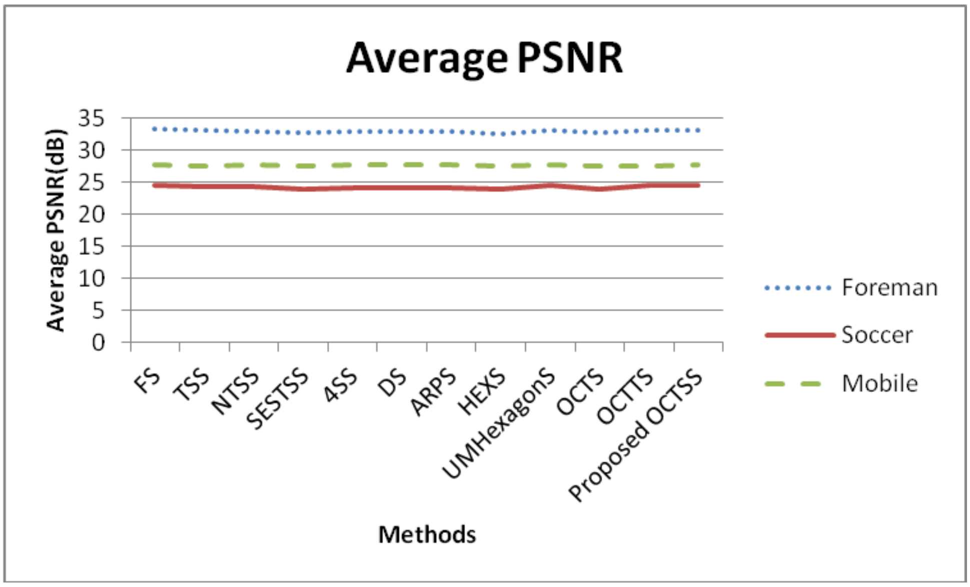


Figure 6. Average PSNR of Various Methods for Mobile, Foreman and Soccer Sequence

From the results, Average PSNR achieved by the proposed work is closer to FS for all three types of sequences and almost higher than other methods. For slow motion sequence, Average PSNR gain is 0.001. There is no gain in other ME algorithms. Number of Search points per macroblock is reduced in the proposed work.

6. Conclusion

Motion estimation is the most vital component of the video compression standard. In this work, combination of octagon and square search pattern is combined in order to find small and large motion vectors. Performance of different Block matching motion estimation algorithms are analyzed and compared with respect to their result of PSNR, MSE and computational complexity. There are many advantages in using this OCTSS method. First, as more number of search points are used in this search in the initial step, the direction of motion is estimated approximately. Hence the first step acheives better PSNR gain. Second, Octagon search pattern uses large step size which is used to identify large motion in the video sequence. Third, square search pattern is used to reduce the number of search points. This reduces computational complexity. Fourth, square pattern is used in the first step to find small motion in the video sequence. Experimental resuls show that the proposed work achieves better PSNR.

From the experimental results, it is proved that OCTS performs better than existing search methods.

References

[1] Y.-C. Lin and S.-C. Tai, “Fast full-search block-matching algorithm for motion-compensated video compression,” IEEE Trans. Commun., **(1997)** Vol. 45, no. 5, pp. 527–531.

[2] R. Li, B. Zeng, and M.L. Liou, “A new three step search algorithm for block motion estimation,” IEEE Trans. Circuits Syst. Video Technol., **(1994)** Vol.4, no.4, pp. 438–442.

[3] 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, April **(1997)** Vol 7, no. 2, pp. 429-433.

[4] L.M. Po and W.C. Ma, “A novel four-step search algorithm for fast blockmotion estimation,” IEEE Trans. Circuits Syst. Video Technol., **(1996)** Vol.6, no.3, pp. 313–317.

[5] M. GHANBARI, “The Cross-Search Algorithm for Motion Estimation” IEEE Trans. on Communications, vol. 38, no. 1, pp. 950-953**,** July **(1990)**

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

[7] C. H. Cheung and L. M. Po, “A novel cross-diamond search algorithm for fast block motion estimation,” IEEE Transactions on Circuits and Systems for Video Technology, December, **(2002)** Vol.12, pp. 1168–1177**.**

[8] C. Zhu, X. Lin, and L. P. Chau, “Hexagon-based search pattern for fast block motion estimation,” IEEE Trans. Circuits Syst. VideoTechnol., May **(2002)** Vol. 12, no. 5, pp. 349–355.

[9] C.-H. Cheung and L.-M. Po, “Novel cross-diamond-hexagonal search algorithms for fast block motion estimation,” IEEE Trans. Multimedia, **(2005)** Vol. 5, no. 1, pp. 16–22.

[10] Xuan Jing and Chau Lap-Pui, "An efficient three-step search algorithm for block motion estimation," IEEE transactions on multimedia, **(2004)** Vol. 6, No.3, pp. 435-438,**.**

[11] L.-P. Chau, C. Zhu, A fast octagon –based search algorithm for motion estimation. J. Signal Process. **(2003)** Vol. 83, 671–675.

[12] Chen, Z., Xu, J., He, Y.Zheng, J, “Fast Integer-pel and fractional-pel motion estimation for H.264/AVC”, Journal of Visual Communication and Image Representation, **(2006)** 17(2), 264-290.

[13] Chunjiang Duanmu, Yu Zhang, “A New Fast Block Motion Algorithm Based on Octagon and Triangle Search Patterns for H.264/AVC”, International Journal of Digital Content Technology and its Applications(JDCTA) June **(2012)** Vol. 6, No. 10.

[14] Bei-li Zou, Cao Shi, Can-Hui Xu, and Shu Chen "Enhanced Hexagonal-Based Search Using Direction-Oriented Inner Search tor Motion Estimation" IEEE transactions on circuits and systems for video technology, Vol. 20, no. I, January **(2010)**.

[15] Kamel Belloulata, Shiping Zhu, and Zaikuo Wang "A Fast Fractal Video Coding Algorithm Using Cross-Hexagon Search for Block Motion Estimation" International Scholarly Research Network ISRN Signal Processing Volume 2011, Article lD 386128, **(2011)**.

[16] M. Ezhilarasan and P. Thambidurai “Simplified Block Matching Algorithm for Fast Motion Estimation in Video Compression” Journal of Computer Science **(2008)** Vol. 4 (4): 282-289.

[17] Shilpa Metkar, Sanjay Talbar, “Motion Estimation Techniques for Digital Video Coding” Springer Briefs in Applied Sciences and Technology, Computational Intelligence, **(2013)**.

[18] Jong-Nam Kim and Tae-Sun Choi, *“A Fast Three Step Search Algorithm with Minimum Checking Points,”* Proc. of IEEE conference on Consumer Electronics, **(1998)** pp.132-133, 2-4.

[19] Faizul Haldi Jamil, Ali Chekima, Rosalyn R, Porle, Othman Ahmad "BMA Performance of video coding for motion estimation" IEEE transaction international conference on intelligent system modelling and simulation (**2012).**

Authors

S. Sowmyayani, she is doing her Ph.D in Computer Science in Manonmaniam Sundaranar University, Tirunelveli, India. She received her M.Sc Degree in Computer Science from St.Xaviers College (Autonomous), Tirunelveli, India in 2011 and M.Phil in Computer Science in Manomaniam Sundaranar University, Tirunelveli, India in 2013. Her research interest includes video compression.

Dr.P.Arockia Jansi Rani B.E, M.E, PhD, she is an Assistant Professor in the department of Computer Science and Engineering in Manonmanium Sundaranar University. She has more than 12 years of Teaching Experience Her research interests include DataMiningand Digital Image Processing.