An Overview of Block Matching Algorithms for Motion Vector Estimation

Conference Paper · June 2017

DOI: 10.15439/2017R85

4 authors, including:

Shailesh Kamble
Yeshwantrao Chavan College of Engineering
28 PUBLICATIONS **54** CITATIONS

SEE PROFILE

Nileshsingh V. Thakur
Nagpur Institute of Technology, Nagpur, India
77 PUBLICATIONS **209** CITATIONS

SEE PROFILE

Proceedings of the Second International Conference on Research in Intelligent and Computing in Engineering pp. 217–222 DOI: 10.15439/2017R85
ACSIS, Vol. 10 ISSN 2300-5963

An Overview of Block Matching Algorithms for Motion Vector Estimation

Sonam T. Khawase[1], Shailesh D. Kamble[2], Nileshsingh V. Thakur[3], Akshay S. Patharkar[4]

[1]PG Scholar, Computer Science & Engineering, Yeshwantrao Chavan College of Engineering, India
[2]Computer Science & Engineering, Yeshwantrao Chavan College of Engineering, India
[3]Computer Science & Engineering, Prof Ram Meghe College of Engineering & Management, India
[4]Computer Technology, K.D.K. College of Engineering, India
[1]sonamkhawase41@gmail.com, [2]shailesh_2kin@rediffmail.com,[3]thakurnisvis@rediffmail.com, [4]akshay.patharkar7@gmail.com

***Abstract*–In video compression technique, motion estimation is one of the key components because of its high computation complexity involves in finding the motion vectors (MV) between the frames. The purpose of motion estimation is to reduce the storage space, bandwidth and transmission cost for transmission of video in many multimedia service applications by reducing the temporal redundancies while maintaining a good quality of the video. There are many motion estimation algorithms, but there is a trade-off between algorithms accuracy and speed. Among all of these, block-based motion estimation algorithms are most robust and versatile. In motion estimation, a variety of fast block based matching algorithms has been proposed to address the issues such as reducing the number of search/checkpoints, computational cost, and complexities etc. Due to its simplicity, the block-based technique is most popular. Motion estimation is only known for video coding process but for solving real life applications many researchers from the different domain are attracted towards block matching algorithms for motion vector estimation.This paper is a review of various block matching algorithms based on shapes and patterns as well as block matching criteria used for motion estimation.**

Keyword- Motion Estimation; Compression; Redundancies; Block Matching Algorithm; Inter-frame; Motion Vectors.

I. INTRODUCTION

A video consists of a sequence of frames and the approach to compressing the sequence of frames can be viewed through two windows i.e. Inter-frame compression and Intra-frame compression. Classification of the frame is shown in Figure 1.Video compression [1,2] is a technology that reduces the bandwidth requirement for transmission of video signals. Lossless and lossy compression are two types of video compression techniques. There is no loss of information in lossless compression and in lossy compression, some amount of information is a loss. In the entire compression process, most computationally expensive and resource hungry operation is motion estimation. Two types of redundancies present in video sequences are spatial and temporal

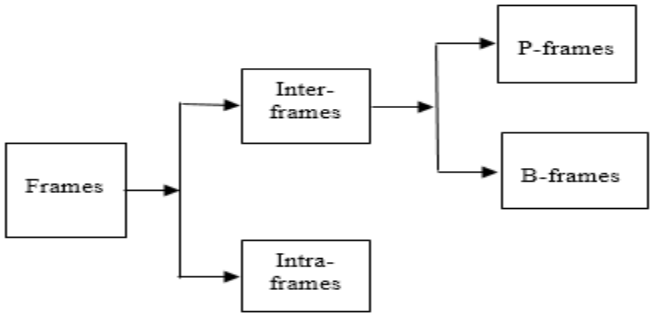

Fig. 1. Classification of Frames

redundancies. Redundancy among neighboring pixel in the image is called spatial redundancy where as redundancy between adjacent frames i.e. current and the reference frame in the sequence of the images is called temporal redundancy [3]. A coding technique that reduces spatial redundancies and temporal redundancies are called intra-frame coding and inter-frame coding respectively. The block schematic of coding technique (intra-frame and inter-frame)is shown in Figure 2. Motion estimation is the process which reduces the temporal redundancies in video sequences. Motion compensation is an algorithm used in the encoding/decoding of video data for video compression. Motion estimation is the most popular technique are used for implementing various video coding standards such as moving picture experts group MPEG-1 to MPEG-4, H.263 and recent/advanced video coding standards H.264/AVC.

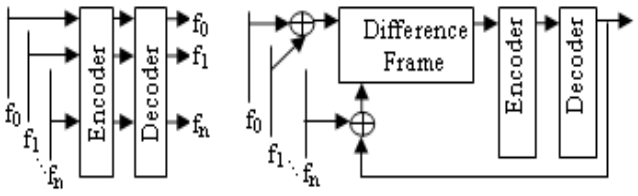

Fig.2. Intra-frame and Inter-frame Coding Technique [6].

This paper is organized as follows: Section II gives block matching criteria for motion estimation. Overview of motion estimation process is given in section III. Related work on block matching algorithms is given in section IV. Section V summarizes the conclusion and future scope from the studied literature followed by references.

II. BLOCK MATCHING CRITERIA

Block matching criteria give a way to find the best match of current block within the search window in the reference frame. Three popular matching criteria used as a block distortion measure (BDM) for block matching motion estimation [4] are mean square error (MSE), mean absolute difference (MAD), Peak to Signal Noise Ratio (PSNR) and sum of absolute difference (SAD) given by equation (1,2,3 and 4) respectively.

$$MSE(i,j) = \frac{1}{N^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\left(C_{ij} - R_{ij}\right)^2 \quad (1)$$

$$MAD(i,j) = \frac{1}{N^2}\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\left|C_{ij} - R_{ij}\right| \quad (2)$$

$$PSNR = 10\log_{10}\frac{(peak\ to\ peak\ value\ of\ data)^2}{MSE} \quad (3)$$

$$SAD(i,j) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\left|C_{ij} - R_{ij}\right| \quad (4)$$

Where N×N is the size of macro block. C_{ij} and R_{ij} are pixels value compared to current and reference macro block respectively. Among all these MAD, MSE is mostly used and PSNR is used for finding image quality. In motion estimation algorithm, the macro block size is an important factor for motion vector computation. In most coding standards, smaller macro block size i.e. 8×8 / 16×16 is used to compute motion vectors (MV) between two macro blocks.

III. OVERVIEW OF MOTION ESTIMATION

The motion estimation is mainly of two types that are forward motion estimation and backward motion estimation. In the backward motion estimation, the candidate frame is considered as current frame & reference frame is the previous frame on which motion vectors are computed. Backward motion estimation escorts to forward motion estimation. Forward motion estimation is opposite of backward motion estimation. In the forward motion estimation, the forward motion vectors are carried out on a frame that appears after the candidate frame. Forward motion estimation drives to backward motion estimation. Motion estimation algorithms are of three types: pixel based, region based and block based. Pixel based algorithm is not preferred due to dependency on the threshold, as threshold value varies from pixel to pixel. Region based is restricted as it has large computations and complexity involved in segmentations. So mostly used is block based algorithm as it is simple and regular. In block matching process [5], the motion vectors are computed between the two frames i.e. current frame and the reference frame. The current frame is divided into a non-overlapping matrix of macro blocks. These macro blocks are compared with a corresponding block and its adjacent neighbors in the reference frame to find motion vector (MV). A motion vector is computed by finding the best suitable block matched between the current frame and the reference frame i.e. f and f+1 respectively. A motion vector (MV) for block B(x, y) is given as (+1, -1) as shown in Figure 3. The MV for all macro blocks comprising a frame gives motion estimated in the current frame. The displacement of pixels position of the reference macro block to the target macro block is called a motion vector (MV).

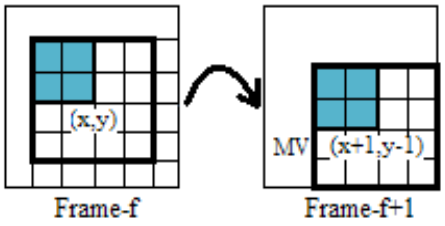

Fig.3. Motion Vector Estimation [6]

A searching window as shown in Figure 4 consists of 'p' pixels on all four sides of the macro block. Where p is called search parameter, as p increases the cost of motion estimation process also increases, p and cost are directly proportional. A macro block is considered as a square of 16×16 pixels and p has a value of 7 pixels in both direction i.e. horizontal and vertical. In the searchwindow, if the size of the block is increased then a total number of blocks that needs to be processed will decrease in each frame. This results in reducing the computational complexity.

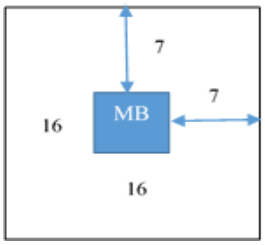

Fig.4. Search Window

IV. RELATED WORK ON MOTION ESTIMATION

The literature associated with block-based motion estimation techniques using various shapes are reviewed and discussed in this section.

A. Rectangular Shape Pattern

Koga and Iinuma [7] proposed the three-step search (TSS) motion estimation algorithm. It employs rectangular search patterns with different sizes. This algorithm is robust and very simple. The TSS algorithm is well known for low bit rate video application. It performs the coarse to fine search method and it gives the motion vectors. The algorithm steps are as follows: Firstly, choose the window size as specified area for searching the best match. Initially, plots eight points around the center point at an appropriate distance of step size and make choice for comparisons. In the step second, if minimum cost point i.e. BDM is found at any one of the nine points then halved the step size and then the center is shifted to that point for next comparison in the third step. In the step third, the second step is repeated until the step size for the window is smaller than one. The TSS requires 25 search/check points. This proposed algorithm doesn't work well on most real-world video sequences in which the motion vector distribution was non-uniformly biased toward the zero vectors.

Li et al. [8] presented the new three-step search (NTSS). NTSS suits well for searching large motion. It is faster and yields better motion estimation as compared to the TSS. The NTSS apply center-biased checking pattern and also consists of halfway-stop technique like TSS used to reduce the computational cost. In the first step of NTSS choose appropriate window size and plot 17 points on selected size of window. Out of these 17 points plot 9 points on the 3 x 3 grid on the central part of the search window and remaining eight points plot on the 9x9 window. If the minimum BDM is found at the center then search stopped/halt, otherwise, go to step 2. In the second step, if one the central neighboring point on the3x3 pattern is found to be minimum BDM point then go to step 3, otherwise goto step 4. In step 3, if the minimum BDM point is found in the middle of a vertical or horizontal edge of 3x3 window then plot 3 points or if the corner of the3x3 grid then plots 5 points. In step 4, if minimum BDM point is found on the outer9x9 window then the halfway-stop technique is used same as in the TSS. In NTSS 17 search/check points are required for the best case, in average case for stationary block 20 or 22 points are required and in the worstcase, 33 points are required.

Jing and Lap-Pui [9] suggested an efficient three-step-search (ETSS) algorithm. ETSS is the modification of three step search method. In ETSS, the motion vector is obtained by applying the small diamond shape pattern in the initial step with unrestricted search steps for searching the center-biased motion vectors. ETSS is useful for the large range of video application and for video sequences which involved complex and large motion such as a movie, sports etc. In ETSS choose search window of size 7. Initially, plot the 13 points. Plot 9 points on a 9x9 pattern same as in TSS and remaining 4 points plot in a pattern of diamond shape around the central point. If the minimum cost point i.e. BDM is obtained at the center then the search will be stopped/halt. If the minimum cost point i.e. BDM is found at the outer eight points then the search is same as TSS algorithm, otherwise goto step 2. In step 2, if minimum cost point i.e. BDM is on the small diamond pattern then additional 3 points will be checked. The center will be changed until the minimum cost point is found to be at the center of the small diamond shape pattern or diamond shape pattern reaches to search window boundary. In ETSS, 13 search/check points needed for the stationary block and in worst case29 points are required. Simulation results show that ETSS algorithm performs better than NTSS in terms of mean square error (MSE) and computation time is saving up to 15% on average.

Po and Ma [10] proposed the four-step search (FSS) algorithm which reduces the search points. It performs better than the TSS and similar to NTSS. In FSS, the center-biased approach is utilized with 9 checking points. In this algorithm the window size 5x5 is used in step 1 then window size is variable it depends on according to the situation which gives the minimum BDM point location. If minimum cost point i.e. BDM is found at the center then goto step 4, otherwise, goto next step. In step 2,if the minimum cost point in step 1 is found at the corner position then, 5 additional checking points are checked and if the cost point found in the previous step at the middle of vertical and horizontal axis then additional 3 search/check points are checked. If the minimum cost point is found at the center then goto step 4, otherwise go to step 3. In step3, searching strategy is same as step 2. In step 4, reduces the window size5x5 to 3x3. FSS requires 17search/check points for best case and 27 search/check points for the worst case. Xu et al. [11] presented improved three step search (ITSS) which is used for low bit-rate video coding applications. The number of computation is reduced by using 11 × 11 search window size instead of 15×15 search window size. It provides better performance than TSS and NTSS.

Nisarand Choi [12] proposed advanced center biased TSS. It reduces the number of computation and improved error performance. Block matching process is speed up by using half way stop technique and it performs better than TSS. Verma et al. [13] suggested fast three step search (FTSS) which is much robust, less computational complexity and has less error. A simple and efficient search (SES) algorithm is proposed by Lu and Liou [14] for motion estimation. The SES and TSS have similar regularity and parallelism but SES has twice sped up factor than TSS. SES performs better than TSS in terms of PSNR and MSE. SES is excellent in low bit rate application for implementation of real-time software applications.

B. Diamond Shape Pattern

For fast motion estimation, a new diamond search (DS) algorithm is proposed by Zhu and Ma [15]. Simulation results validate that it performs better than the three-step search (TSS). DS algorithm gives a close performance as new three-step

search (NTSS) but required22% of less computation on average. Two types of patterns are present in DS algorithm i.e. large diamond shape search pattern (LDSP) which consists of nine checking points out of these eight are plotted around the center one. The second pattern is called small diamond shape search pattern (SDSP) it comprises five checking point i.e. small diamond. The algorithm consists of following steps: In step 1, LDSP is plotted on the search region then checks the 9 points of LDSP. If minimum cost point is obtained at the center position then goto step 3, otherwise,goto step 2. In step 2, form new LDSP pattern if minimum BDM point in the next step is obtained at a point other than the center of LDSP. If the new minimum cost point is found at the center position then goto step 3 otherwise repeat step 2. In step 3, shift LDSP pattern to SDSP pattern if minimum cost point found at the center of LDSP. The minimum cost point is obtained in step 3 then calculate MV. DS requires 13 searches/check points in the best case. DS algorithm is better than 4SS and block-based gradient descent search (BBGDS), in terms of MSE and required a number of searching/checking points.

Small cross diamond search (SCDS) proposed by Hongand Po [16], motion estimation process is faster with negligible loss in video quality and it requires a very few number of search/check points. Cheng et al. [17] introduced modified diamond search (MDS) consist of the shape of small diamond search pattern (SDSP) in the initial search step then used a simplified large diamond search pattern (LDSP) in next search. Singh and Ahamed [18] introduced modified small cross diamond search for fast motion estimation. Nie [19] proposed adaptive rood pattern search (ARPS) algorithms, which consists of initial and refined the local search. In video coding applications it is a very robust and efficient motion estimation algorithm. The computation of video coding is greatly reduced with ARPS. Cheung and Po [20] suggested normalized partial distortion search algorithm whose MSE performance is same as full search (FS) algorithm but it reduces the computation 12 to13 times.

C. Hexagonal Shape Pattern

A novel hexagon-based search algorithm (HEXBS)is proposed by Zhu et al.[21]. The HEXBS algorithm calculates motion vector with less number of search points than the diamond search (DS) algorithm. The speedup factor of the HEXBS algorithm is more than the DS algorithm striking for finding large motion vectors. The HEXBS algorithm can discover any motion vector having fewer searches than the DS. Plot 7 points along with center point on the specified search area. If the minimum cost point i.e. BDM is obtained at the center of large hexagon shape pattern (LHSP) then LHSP is shifted to small hexagon shape (SHSP)and final MV is computed from the SHSP point. The small hexagon consists of four points. The final new cost point is the solution for the motion vector. Otherwise, if minimum cost point is other than the center of LHSP then form new LHSP to check another three points and again this process is repeated until minimum cost point found at the center of LHSP. HEXBS requires only 11 searches/check points for the best case. The HEXBS gives the efficient result than the FS, TSS, NTSS and DS. This algorithm shows the substantial speed enhancement and equal distortion performance with the diamond search (DS).

The cross-hexagon-based motion estimation algorithm using motion vector adaptive search technique is proposed by Li Hong-ye et al. [22].The proposed algorithm is combined with hexagon and cross search pattern algorithm. The experimental results show that the speed of the search algorithm is superior and searching points close to Hexagon algorithm (HEXBS).Zhu et al. [23] proposed an enhanced hexagonal search algorithm (EHEXBS) to improve the performance in terms of reducing the number of search/check points and block distortion. The experimental results show that the EHEXBS algorithm outperforms the original HEXBS up to 57% in both speed improvement rate & distortion decrease rate. Chorng-For further reducing the number of searching points Yann Su et al.[24] proposed an inner search algorithm, named efficient hexagonal inner search (EHIS). The EHIS uses the central minimal distortion information and exploits the distortion information of the points located in the hexagonal shape pattern. The EHIS uses the central minimal distortion information and exploits the distortion information of the points located in the hexagonal pattern. Simulation results show that EHIS performs better in terms of the number of search/check points and MSE than EHEXBS for fast block motion estimation. Cheung and Po [25] proposed cross-diamond hexagonal search (CDHS). Before the first step of CDS, CDHS employs a smaller cross-shaped pattern and in next step replaces the diamond-shaped with hexagonal search patterns (HSP).

Belloulata et al. [26] suggested new cross-hexagon search (NHEXS) for block motion estimation for fractal video coding. The NEHEX uses two cross-shapedpatterns as the first two initial step and large small hexagon shaped patterns as the subsequent steps for BME. NHEXS apply half way stop technique to achieve significant speedup on sequences with the stationary and quasi-stationary block. Experimental results indicate that it reduces the encoding time & increases the compression ratio and quality of the video. Kamble et al. [27] proposed modified three-step search (MTSS) which consists of three-step search (TSS) and Hexagonal Search (HEXBS).A new block matching motion estimation approach i.e. diamond-arc-hexagon search (DAHS) which uses diamond, arc, and hexagon search patterns to accomplish the fast searching process proposed by Lin and Chiang [28]. DAHS are best for both quasi-stationary and large motion computation.

D. Motion Estimation for Video Coding Standards

Motion estimation widely used to video coding standards such as MPEG-1 to MPEG-4, H.264 to H.263 and recent/advanced

video coding standards H.264 of video data for purpose of storage requirement and transmission.

Park et al. [29] proposed fast three-step search in H.264. For searching motion vector, it uses error surface and pixel interpolation property of SAD. Time taken to calculate for motion estimation was reduced by approximately 24% for encoder processing and by approximately 51 % for sub pixel processing. Parallel architecture for efficient new three-step search (NTSS) in H.264 is proposed by Ho et al. [30]. Among the all existing previous works, this architecture is the best tradeoff in terms of hardware area overhead and speed. To separately encoded frame, it adopts the partitioning technique. Experimental results show that NTSS architecture can reduce the encoder time while maintaining similar PSNR compared with traditional NTSS.

So and Wu [31] suggested four step genetic search achieving better quality of video. This algorithm is suitable for H.261, MPEG-1, and MPEG-2. Tsai and Pan [32] proposed 3-D predict hexagonal search (3DPHS) on video coding standard H.264. 3DPHS depends on the motion vector distribution characteristics. It uses a rood-shaped search pattern at first two steps of searching with a higher probability to get motion vector. It also predicts the movement of an object in a vertical and horizontal direction. Simulation results show that it increases the speed by 25% to 75% over other fast block matching algorithms. New hardware-oriented modified diamond search for H.264/AVC for motion estimation is proposed by Ndili and Ogunfunmi [33]. The goal of this architecture is to support low bit rate applications as well as of high-quality video mobile devices with low power. This architecture is more hardware-efficient. Simulation results show that, it has better speedup quality and rate-distortion performance. Kamble et al. [34] proposed a modified three-step search (MTSS) block matching motion estimation algorithms and finite automata theory based fractal coding approach.

V. CONCLUSION AND FUTURE SCOPE

Based on studied literature, it is found that there is a scope for improvement on developing the fast motion estimation algorithm in terms of reducing the number of search/check points. The performance of different existing motion estimation algorithms is measured by conducting experimentation on different video sequences i.e. soccer, suzie, traffic, football, ice, akiyo, salesmen, missa, tennis, and mobile etc. within macro block size 8×8 or 16×16. The comparison for different block matching algorithms discussed in this paper based on the number of search/check points per macro block is given in section IV. Existing algorithms for motion vector estimation used in various applications such as medical, security, space science and psychological studied etc. The MAD and MSE are mostly used as a block distortion measures for motion estimation. Among the all motion vector estimation algorithms based on block matching received a high attention by researcher because of their simplicity and regularity.

These motion estimation algorithms are limited by shape pattern and searching speed so that fast movements in video sequences can't find exact/correct motion vectors. There is the scope for expansion in developing the new block matching algorithm or by combining existing algorithms i.e. hybrid algorithm based on different shape and search patterns so the computational cost, the number of the searching point will reduce as compared with traditional fast motion estimation algorithm. Also, we will try to implement new block-matching estimation algorithm for efficient video compression and tracking single/multiple objects in the video. In future, other evolutionary computing techniques i.e. nature-inspired algorithm- genetic algorithm, particle swarm optimization etc. also can be tried to estimate motion vectors for better performance. Comparisons of existing algorithms in terms of the searching points are given in Table I.

TABLE I: COMPARISON IN TERMS OF SEARCHING POINTS

Motion Estimation Algorithms	**Number of Searching Points**		
	Best Case	***Average Case***	***Worst Case***
Full Search (FS)	225	225	225
Three Step Search (TSS)	25	25	25
New Three Step Search (NTSS)	17	17 to 33	33
Efficient Three Step Search (ETSS)	13	13 to 29	29
Improved Three Step Search (ITSS)	17	17 to 22	22
Four Step Search (FSS)	17	17 to 27	27
Diamond Search (DS)	13	13 to 30	30
Cross Diamond Search (CDS)	9	9 to 29	29
Hexagonal Search (HS)	11	11 to 30	30

REFERENCES

[1] S. Kamble, N. Thakur, L. Malik, P. Bajaj, "Color video compression based on fractal coding using quad-tree weighted finite automata," *Information system design and intelligent application, Proceedings of Second International Conference INDIA 2015*, volume 2, advances in intelligent system and computing, Springer India, volume 340, pp. 649-658, 2015.

[2] S. Kamble, N. Thakur, L. Malik, P. Bajaj, "Quad-Tree Partitioning and Extended Weighted Finite Automata Based Fractal Color Video Coding," Int. J. Image Mining, vol. 2, no. 1, pp. 31-56, 2016.

[3] H. Surrah and M. Haque, "A Comparative Approach for Block Matching Algorithms used for Motion Estimation," *IJCSI*, volume 11, no. 3, pp. 562-568, 2014.

[4] E. Chan and S. Panchanathan, "Review of Block Matching Based Motion Estimation Algorithms for Video Compression, *International Journal of Scientific & Engineering Research*, vol. 1, no. 3, pp. 151-153, 1993.

[5] Aw. Hussain, L. Knight, D. Al-Jumeily, P. Fergus, and H. Hamdan, "Block Matching Algorithms for Motion Estimation–A Comparison Study", *International Journal of Scientific & Engineering Research*, vol. 264, no. 2, pp. 356-369, 2014.

[6] S. Kamble, N. Thakur and P. Bajaj, "A Review on Block Matching Motion Estimation and Automata Theory based Approaches for Fractal Coding*", International Journal of Interactive Multimedia and Artifical Intelligence,* volume 4, no.2, pp.91-104, 2016.

[7] T. Koga, K. Iinuma, "Motion Compensated Inter frame Coding for Video Conferencing," *in proceedings of the NTC*, pp. 230-236, 1981.

[8] R. Li, B. Zeng, and M. L. Liou,"A new Three-Step Search Algorithm for Block Motion Estimation,"*IEEE Trans .on Circuits and Systems for Video Technology,*volume 4, no. 4, pp. 438-442.

[9] Xuan Jing and Chau Lap-Pui, "An Efficient New Three-Step Search algorithm for block motion estimation," *IEEE Transactions on multimedia*, vol. 6, no.3, pp. 435-438, 2004.

[10] L. Po and W. Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation,"*IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313-317, 1996.

[11] D. Xu, C. Bailey and R. Sotudeh, "An Improved Three Step Search Block Matching Algorithm for Low Bit Rate Video Coding Applications," *International Symposium on Signals, Systems, & Electronics Conference in proceedings of Pisa,* no. 10,pp. 178-181, 1998.

[12] Humaira Nisar and T. Choi, "An Advanced Center Biased Three Step Search Algorithm for Motion Estimation," *IEEE International Conference on Multimedia and Expo in proceedings of Latest Advances in the Fast Changing World of Multimedia, vol. 1, no. 10, pp. 95-98, 2000.*

[13] N. Verma, T. Sahu and P. Sahu, "Efficient Motion Estimation by Fast Three Step Search Algorithms,"*International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* volume 1, no. 5, pp. 380-385, 2012.

[14] JianhuaLuandMingL.Liou,"ASimple and Efficient Search Algorithm for Block-Matching Motion Estimation," *IEEE Transactions onCircuits and Systems for Video Technology*, volume 7, no. 2, pp. 429-433, 1997.

[15] S. Zhu and K. Ma, "A new Diamond Search Algorithm for Fast Block Matching Motion Estimation," *IEEE Transaction on Image Processing,* vol. 9, no. 2, pp. 287-290, 2000.

[16] C. Hong and L. M. Po, " A Novel Small-Cross-Diamond Search Algorithm for Video Coding and Videoconferencing Applications," *IEEE Trans. on Circuits and Systems for Video Technology*, volume 12, no. 12, pp. 681-684, 2002.

[17] YunCheng,Xine You,MinlianXiao,"A Modified Diamond Search Algorithm," *IEEE International Symposium on IT in Medicine & Education, Cuangzhou,* pp. 481-485, 2011.

[18] K. Singh and S. Ahamed, "Modified Small-Cross Diamond Search Motion Estimation Algorithm for H.264/AVC," *IEEE India Conference,* pp. 1-5, 2013.

[19] Y. Nie, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation," *IEEETransactions on Image Processing*, vol. 11, no. 12, pp. 1442-1448, 2002.

[20] C. Cheung and L. Po, "Normalized Partial Distortion Search Algorithm for Block Motion Estimation," *IEEE Trans. on Circuits and Systems for Video Technology,* volume 10, no. 3, pp. 417-422, 2000.

[21] Ce Zhu, Xiao Lin , Lap-Pui Chau ,Keng-PangLim , Hock-Ann Ang Choo-Yin Ong, "A novel hexagon-based search algorithm for fast block motion estimation" *IEEE International Conference Acoustics, Speech, and Signal Processing Proceedings*. (ICASSP '01), 2001.

[22] Li Hong-ye "Cross-Hexagon-based motion estimation algorithm using motion vector adaptive search technique," *IEEE transaction* 978-1-4244-5668, 2009.

[23] CE Zhu, Xiao Lin, Lappui Chau &Lai Man Po "Enhanced hexagonal search for fast block motion Estimation," *IEEE Transaction on circuits &system for video technology* volume 14, no. 10, 2004.

[24] Chorng-Yann Su, Yi-Pin Hsu, and Cheng-Tao Chang "Efficient Hexagonal Inner Search For Fast Motion Estimation*" IEEE transaction on multimedia*7803-9134-9 volume 4, 2005.

[25] Chun-Ho Cheung, Lai-Man Po, "Novel Cross-Diamond-Hexagonal Search Algorithms for Fast Block Motion Estimation," *IEEE Transactions on Multimedia,* volume 7, no. 1, February, pp. 16-22, 2005.

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

[27] S. Kamble, N. Thakur, L. Malik and P. Bajaj, "Fractal Video Coding using Modified Three-Step Search Algorithm for Block Matching Motion Estimation*," Computational Vision & Robotics in Proceedings of International Conference on Computer Vision & Robotics, Advance in Intelligent Systems & Computing,* volume 332, pp. 151-162, 2015.

[28] Han–Ting Lin & Jen Shiun Chiang "A new diamond Arch- Hexagon Search algorithm for fast block motion estimation " *IEEE International symposiumon signal processing and information Technology,*pp.811-816, 2006.

[29] Dongkyun Park, Young Jang, and JongHwa Lee, "A new Fast Three Step Search Motion Estimation Algorithm in H.264,"*IEEE transactions on circuits and systems for video technology,*978-1-4244-3589 pp. 541-544, 2007.

[30] Mean-Hom Ho, Jan-Jun Huang, Shang-Chiang Chin and Chun-Lung Hsu "High Efficient NTSS-Based Parallel Architecture for Motion Estimation in H.264"*IEEE transaction*978-1-4244-2064, vol.3, pp. 679-683, 2008.

[31] Man F. So and Angus Wu "Four-Step Genetic Search for Block Motion Estimation"*IEEE transaction*0-7803-4428-6138, pp.1393-1396, 1998.

[32] Tsung-Han Tsai and Yu-Nan Pan, "A Novel 3-D Predict Hexagon Search Algorithm for Fast Block Motion Estimation on H.264 Video Coding*," IEEE transactions on circuits and systems for video technology,* volume 16, no. 12, pp.1542 -1547, 2006.

[33] Obianuju Ndili and Tokunbo Ogunfunmi "Algorithm and Architecture Co-Design of Hardware-Oriented, Modified Diamond Search for Fast Motion Estimation in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, volume 21, no. 9, 2011.

[34] S. Kamble, N. Thakur and P. Bajaj, "Modified Three-Step Block Matching Motion Estimation and Weighted Automata based Fractal video Compression*", International Journal of Interactive Multimedia and Artifical Intelligence,* vol. 4, no.4, pp.27-39, 2017.