Darshna D.Jagiwala, Prof. Mrs. S. N. Shah/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 6, November- December 2012, pp.1396-1401

Analysis of Block Matching Algorithms for Motion Estimation in H.264 Video CODEC

Darshna D.Jagiwala, Prof. Mrs. S. N. Shah
M.Tech. (Research),
ECED, SVNIT,Surat,Gujarat(INDIA),
Assistant Professor,
ECED, SVNIT,Surat,Gujarat(INDIA),

ABSTRACT

H.264 video compression standard has already been introduced in new electronic gadgets such as mobile phones and digital video players, and has gained fast acceptance by end users. Service providers such as online video storage and telecommunications companies are also beginning to adopt H.264 standard. The key element to high performance of video compression lies in an efficient reduction of the temporal redundancy. For this purpose, block matching algorithms used for motion estimation technique has been successfully applied in the video compression standards from MPEG1 / H.261 to MPEG4 / H.264. This paper is an analysis of the block matching algorithms used for motion estimation in H.264 video compression standard. It implements and compares 5 different types of block matching algorithms which are Exhaustive Search, Three Step Search, Diamond Search, Four Step Search and Adaptive Road Pattern Search on H.264 Video codec.

Keywords-H.264 video CODEC, Block matching algorithms, Motion estimation, PSNR

I.INTRODUCTION

Video Compression is essential for Multi-Media Communication. Video compression is the process of reducing the amount of data required to represent a digital video signal, prior to transmission or storage. Video compression (or video coding) is an essential technology for applications such as digital television, DVD-Video, mobile TV, videoconferencing and internet video streaming. The complementary operation, decompression or decoding, recovers a digital video signal from a compressed representation, prior to display. Since the early 1990s, when video coding technology was in its infancy, international standards such as H.261, MPEG-1, H.262/MPEG-2 Video, H.263, and MPEG- 4 Part 2 have been powerful engines behind the commercial success of digital video. They have played a pivotal role in establishing the technology by ensuring interoperability among products developed by different manufacturers. Moreover, these standards have permitted economies of scale to allow steep reductions in cost for mass-market affordability.

The latest addition to the lineup of these well-known standards is H.264/AVC. . H.264 is the result of a joint project between the ITU-T's Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group (MPEG). ITU-T is the sector that coordinates telecommunications standards on behalf of the International Telecommunication Union. ISO stands for International Organization for Standardization and IEC stands for International Electrotechnical Commission, which oversees standards for all electrical, electronic and related technologies [1].

Block matching motion estimation was one of the most important modules in the design of any video encoder[2].In motion estimation process, firstly, a frame is partitioned into many non overlapping blocks with different block sizes according to the motion contents, secondly, the target block in the current frame is compared with the candidate blocks in the reference frame; under the constraint of cost function, we can obtain the best matched block by minimizing the cost function, and finally, the motion vector(MV), which represents the displacement between the current block and the best matched block, together with the residual signal, which is the pixel difference between the current block and the best matched block, are transported to the next process to be coded. Motion estimation is quite computationally intensive and can consume up to 80% of the computational power of the encoder if the full search (FS) is used by exhaustively evaluating all possible candidate blocks within the search window. Therefore, fast algorithms are highly desired to significantly speed up the process without sacrificing the distortion seriously. Many computationally variants [3]-[7] were developed for block based motion estimation. In this paper, the fast motion estimation algorithms which are Exhaustive Search (ES), Three Step Search (TSS), Diamond Search (DS), Four Step Search (4SS) and Adaptive Road Pattern Search (ARPS) applied on H.264 CODEC.

The rest of this paper is organized as follows: the H.264 encoder and the concept of fast ME algorithms has been reviewed separately in section II. Section III compares them on H.264 CODEC and

presents some simulation results, followed by conclusion and references.

II.H.264 CODEC AND CONCEPT OF BLOCK MATCHING ALGORITHMS

The H.264/AVC design consists of a network abstraction layer (NAL) and a video coding layer (VCL). The NAL, which was created to fulfill the network-friendly design objective, formats data and provides header information for conveyance by transport layers or storage media. The VCL is specified to efficiently represent the content of the video data and fulfill the design objective of enhanced coding efficiency. The generalized block diagram of typical encoder processing elements for the VCL is provided in Figure 1.

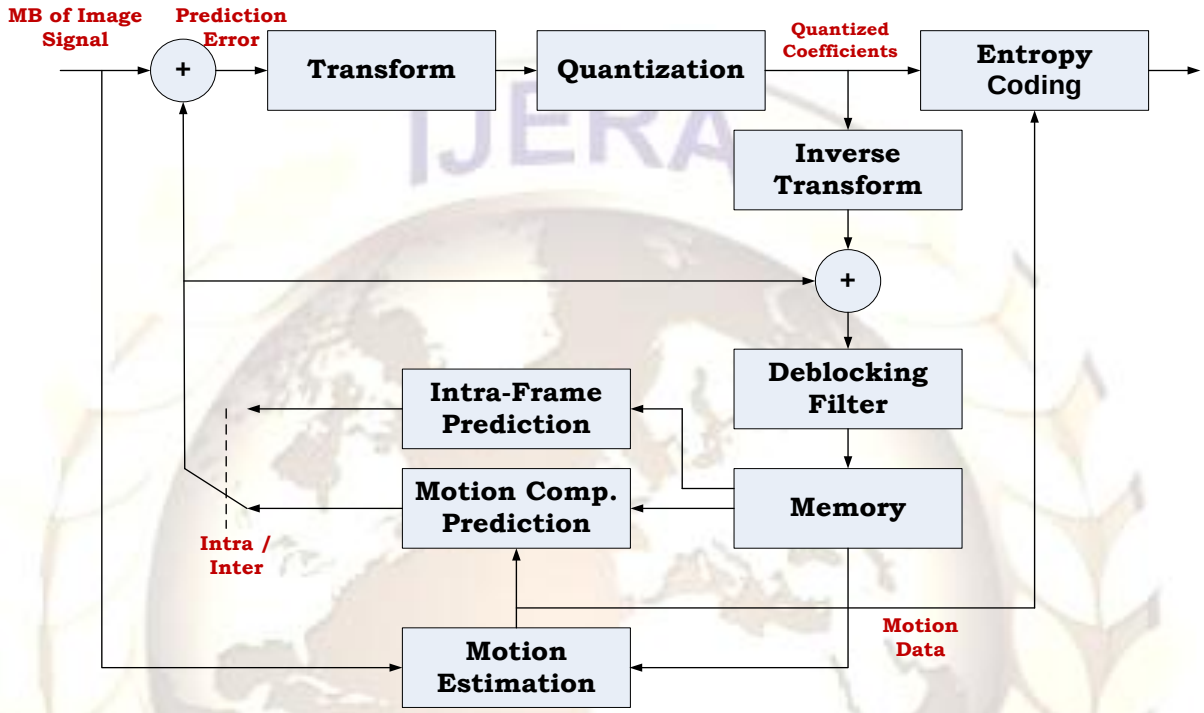

Fig.1 Generalized Block Diagram of a Hybrid Video Encoder

Decoding processes are conceptually a subset of these encoding processes [8]. Two types of redundancy are responsible for compression. Redundancy between the pixels called spatial redundancy. Redundancy between frames called temporal redundancy. Spatial redundancy, also known as Intra-frame redundancy is achieved using techniques of Integer Transform, Quantization and Entropy coding. Temporal redundancy, also known as Inter-frame redundancy is achieved using techniques of motion estimation. Block based motion estimation is accepted in all the video coding standards proposed till date. It is easy to implement in hardware and real time motion estimation and prediction is possible. If a block of N × N pixels from the candidates frame at the coordinate position (r, s) is considered and then consider a search window having a range ±w in both end directions in the references frame, as shown in Figure 2. For each of the $(2p+1)^2$ search position (including the current row and the current column of the reference frame), the candidate block is compared with a block of size N × N pixels, and the best matching block, along with the motion vector is determined only after all the $(2p+1)^2$ search positions are exhaustively explored. By exhaustively testing all the candidate blocks within the search window, this algorithm gives the global minimum block distortion position which corresponds to the best matching block.

The search area for a good macro block match is constrained up to *p* pixels on all fours sides of the corresponding macro block in previous frame. This '*p*' is called as the search parameter. Larger motions require a larger *p*, and the larger the search parameter the more computationally expensive the process of motion estimation becomes. Usually the macro block is taken as a square of side 16 pixels, and the search parameter *p* is 7 pixels. The matching of one macro block with another is based on the output of a cost function. The macro block that results in the least cost is the one that matches the closest to current block. There are various cost functions, of which the most popular and less computationally expensive is Mean Absolute Difference (MAD) given by equation (i). Another cost function is Mean Squared Error (MSE) given

by equation (ii) [9]. Consequently, many algorithms with a reduced number of search locations have been proposed, including the Exhaustive Search (ES) [10], Three Step Search (TSS) [11], Diamond Search (DS) [12], Four Step Search (4SS) [13] and Adaptive Road Pattern Search (ARPS) [14].

$$MAD = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|Cij - Rij| \quad (i)$$

$$MSE = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(Cij - Rij)^2 \quad (ii)$$

Where N is the side of the macro bock, Cij and Rij are the pixels being compared in current macro block and reference macro block, respectively.

Peak-Signal-to-Noise-Ratio (PSNR) given by equation (iii) characterizes the motion compensated image that is created by using motion vectors and macro blocks from the reference frame. Based on this theory some blocked based ME algorithms are explained below:

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

A. Exhaustive Search (ES): This algorithm is the most computationally expensive block matching algorithm of all. It is also known as Full search algorithm. This algorithm calculates the cost function at each possible location in the search window. As a result it finds the best possible match and gives the highest PSNR amongst any block matching algorithm.

B. Three Step Search (TSS): It became very popular because of its simplicity and also robust and near optimal performance

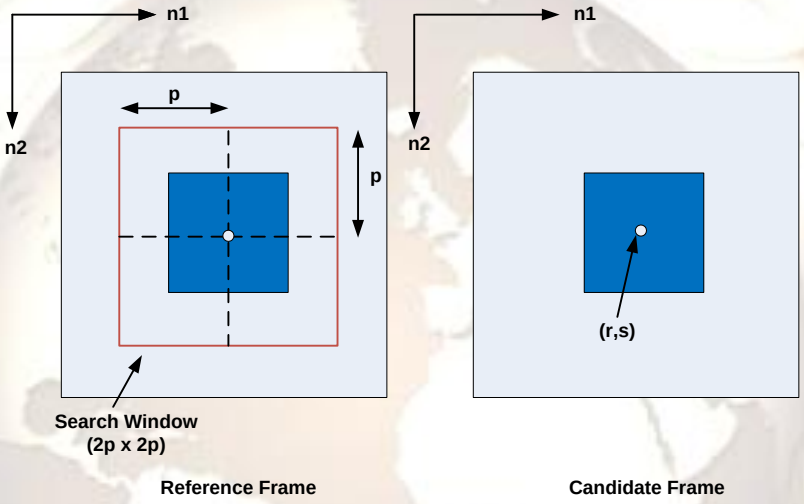

Fig.2 Exhaustive (Full) Search Motion Estimation

It searches for the best motion vectors in a course to fine search pattern. The algorithm may be described as:

Step 1: An initial step size is picked. Eight blocks at a distance of step size from the centre (around the centre block) are picked for comparison.

Step 2: The step size is halved. The centre is moved to the point with the minimum distortion. The point which gives the smallest criterion value among all tested points is selected as the final motion vector m.TSS reduces radically the number of candidate vectors to test, but the amount of computation required for evaluating the matching criterion value for each vector stays the same.

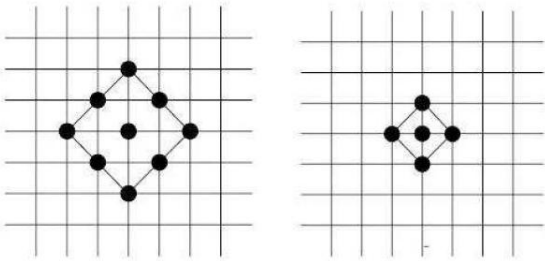

Fig. 3. (a) Large Diamond Search Pattern (LDSP), (b) Small Diamond Search Pattern (SDSP)

C. Four Step Search (4SS): The 4SS algorithm [utilizes a center-biased search pattern with nine checking points on a 5 x 5 window in the instead of a 9 x 9 window in the TSS. This algorithm helps in reducing the number of search points when compared to the TSS and hence is more robust. The 4SS algorithm is summarized as follows:

Step 1: A minimum Block Distortion Method (BDM) point is found from a nine-checking point's pattern on a 5 x 5 window located at the center of the 15 x 15 searching area as shown in Figure. 4(a). If the minimum BDM point is found at the center of the search window, go to Step 4; otherwise go to Step 2: The search window size is maintained in 5 *x* 5. However, the search pattern will depend on the position of the previous minimum BDM point.

a) If the previous minimum BDM point is located at the corner of the previous search window as shown in Figure. 4(b), five additional checking points are used.

b) If the previous minimum BDM point is located at the middle of horizontal or vertical axis of the previous search window, three additional checking points are used as shown in Figure. 4(c). If the minimum BDM point is found at the center of the search window, go to Step 4; otherwise go to Step 3.

Step 3: The searching pattern strategy is the same as Step 2, but finally it will go to Step 4.

Step 4: The search window is reduced to 3 x 3 as shown in Figure. 4(d) and the direction of the overall motion vector is considered as the minimum BDM point among these nine searching points.

From the algorithm, we can find that the intermediate steps of the 4SS may be skipped and then jumped to the final step with a 3 x 3 window if at any time the minimum BDM point is located at the center of the search window. Based on this four-step search pattern, we can cover the whole 15 x 15 displacement window even only small search windows, 5 x 5 and 3 x 3, are used. This 4SS produce better performance than the TSS and it also possesses the regularity and simplicity of hardware-oriented features.

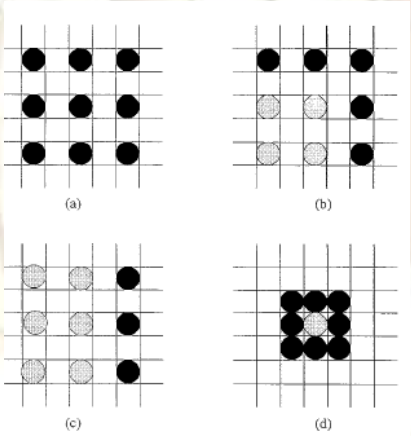

Fig. 4. Search Patterns Of 4SS. (a) First Step, (b) Second/Third Step, (c) Second/Third Step, (d) Fourth Step

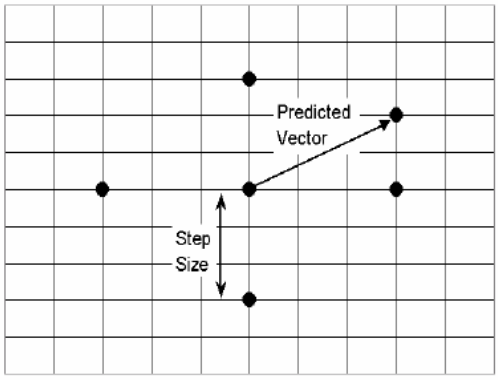

Fig. 5 Adaptive Root Pattern: The predicted motion vector is (3,-2) ,and the step size S = Max(|3|, |-2|) = 3

D. Diamond Search (DS): The DS algorithm employs two search patterns. The first pattern, called Large Diamond Search Pattern (LDSP) as shown in Figure.5 (a) comprises nine checking points from which eight points surround the center one to compose a diamond shape. The second pattern consisting of five checking points forms a small diamond shape, called Small Diamond Search Pattern (SDSP) as shown in Figure. 5 (b). The DS algorithm is summarized as follows.

Step1: The initial LDSP is centered at the origin of the search window, and the 9 checking points of LDSP are tested. If the MBD point calculated is located at the center position, go to Step 3; otherwise, go to Step 2.

Step2: The MBD point found in the previous search step is re-positioned as the center point to form a new LDSP. If the new MBD point obtained is located at the center position, go to Step 3; otherwise, recursively repeat this step.

Step3: Switch the search pattern from LDSP to SDSP. The MBD point found in this step is the final solution of the motion vector which points to the best matching block.

This algorithm consistently performs well for the image sequence with wide range of motion content. It also outperforms the well-known TSS algorithm while reducing computation by 20%-25% approximately.

E. Adaptive Rood Pattern Search (ARPS): This algorithm makes use of the fact that the general motion in a frame is usually coherent, i.e. if the macro blocks around the current macro block moved in a particular direction then there is a high probability that the current macro block will also have a similar motion vector. This algorithm uses the motion vector of the macro block to its immediate left to predict its own motion vector. An example is shown in Fig 5. The predicted motion vector points to (3, -2). In addition to checking the location pointed by the predicted motion vector, it also checks at a rood pattern distributed points, where they are at a step size of S = Max (|X|, |Y|). X and Y are the x-coordinate and y-coordinate of the predicted motion vector. This rood pattern search is always the first step. It directly puts the search in an area where there is a high probability of finding a good matching block. The point that has the least weight becomes the origin for subsequent search steps, and the search pattern is changed to SDSP. The procedure keeps on doing SDSP until least weighted point is found to be at the center of the SDSP. A further small improvement in the algorithm can be to check for Zero Motion Prejudgment, using which the search is stopped half way if the least weighted point is already at the center of the rood pattern.

III. SIMULATION RESULTS

In this work, three video sequences news, shakycar & akiyo are used for compression. News & akiyo are in QCIF format (176 X 144) at 25 FPS frame rate & shakycar sequence is in CIF format (320X240) at 25 FPS frame rate. The coding performances are taken based on 100 no. of frames and PSNR of the encoded video sequences. Encoded P frames & B frames are used for PSNR computation. In the tests, IBBPBBPBBI GOP structure and QP (Quantization Parameter) = 10 are taken for encoding purpose. Here all graphs are generated in MATLAB software.

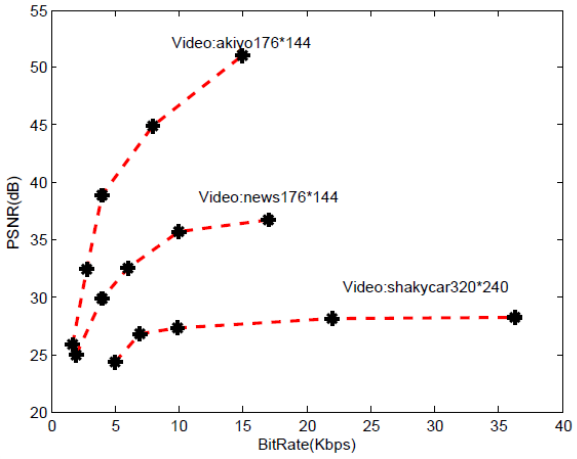

Fig.6 BitRate-PSNR curves of All Video sequences encoded using H.264 encoder

In H.264 simulations, constant quantization parameter values of 10, 20, 30, 40, and 50 were used to cover a practical range of low to moderately high bit rates. Here for each sequence, bitrate-quality curves are produced. Figure. 6 shows the bitrate-quality curves for all video sequences. The PSNR comparison of the compensated images generated using the motion estimation algorithms is shown in Figure 7,8,9 for news, akiyo and shakycar respectively. Also performance comparison of computations for all video sequences is included in Table 1. The results are similar to the results of [9].As shown in Figure 7, 8and 9,PSNR value of TSS and DS are approximately close to each other, but the computations of TSS is greater than DS for all the video sequences. Here ES takes more no. of searches per MB, but gives highest value of PSNR in H.264 CODEC.PSNR value of 4SS is more than that of ARPS while FSS requires more no. of computations as compared to ARPS for all sequences.

Table 1. Performance comparison of computations for all video sequences

Video Sequence	ES Computations	TSS Computations	4SS Computations	DS Computations	ARPS Computations
News	210.1010	24.1205	16.6775	13.3754	5.0982
Akiyo	210.1010	24.1042	16.4400	12.6768	5.0720
shakycar	216.3333	24.5912	23.0731	19.8675	11.6631

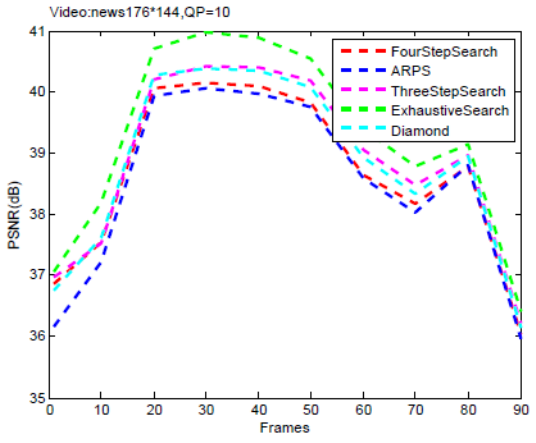

Fig.7 PSNR Performance of Fast Block Matching Algorithms for news Sequence using H.264 encoder

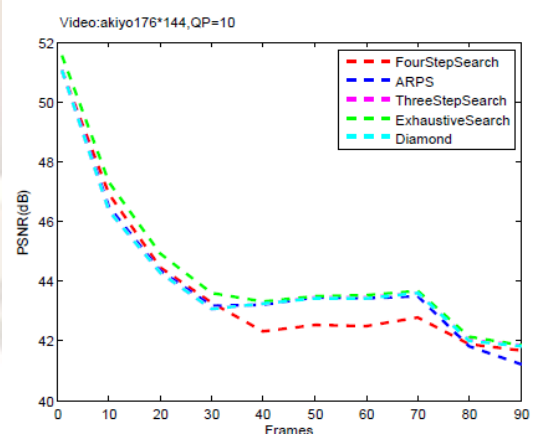

Fig.8 PSNR Performance of Fast Block Matching Algorithms for akiyo Sequence using H.264 encoder

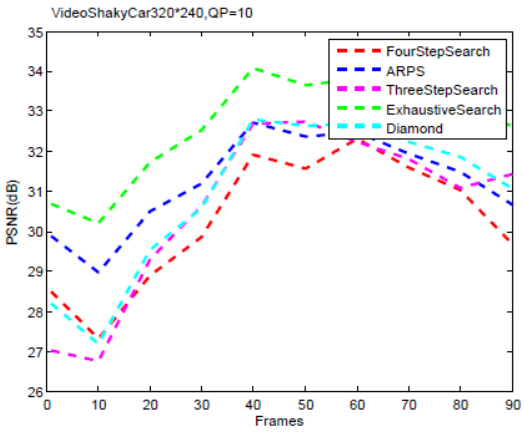

Fig.9 PSNR Performance of Fast Block Matching Algorithms for shakycar Sequence using H.264 encoder

IV. CONCLUSION

In this paper, an overview of some Block matching motion estimation algorithms range from the very basic Full Search to the recent fast adaptive algorithms like Pattern Based Search in H.264 CODEC has been discussed. Full search Motion Estimation algorithm is not fit for real-time applications because of its unacceptable computational cost. Normally ME is quite computationally intensive and can consume up to 80% of the computational power of the encoder if the full search is used by exhaustively evaluating all possible candidate blocks within the search window. As a consequence, the computation of H.264 video coding is greatly reduced with pattern based block motion estimation.

REFERENCES

[1] Advanced Video Coding for Generic Audiovisual Services,ITU-T Rec.H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005, Version 4: Sept. 2005, Version 5 and Version 6: June 2006, Version 7: Apr. 2007, Version 8 (including SVC extension): Consented in July 2007.

[2] Xie Liyin, Su Xiuqin, Zhang Shun , "A Review of Motion Estimation Algorithms for Video Compression",2010 International Conference on Computer Application and System Modeling (ICCASM 2010) 978-1-4244-7237-2010 IEEE.

[3] M. Ghanbari, "The cross-search algorithm for motion estimation," IEEE Trans.Commun., vol. 38, pp. 950-953, July 1990.

[4] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438442, Aug.1994.

[5] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation,"IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313317, June 1996.

[6] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," in Proc. Int. Conf. Inf. Commun. Signal Process. (ICICS '97), vol. 1, Sep. 9–12, 1997, pp.292–296.

[7] M. F. So and A. Wu, "Four-step genetic search for block motion estimation," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process.(ICASSP '98), vol. 3. May 1998, pp.1393–1396.

[8] Jörn Ostermann, Jan Bormans, Peter List,Detlev Marpe, Matthias Narroschke,Fernando Pereira, Thomas Stockhammer, and Thomas Wedi, "Video coding with H.264/AVC: Tools, Performance, and Complexity",IEEE Circuits And Systems Magazine First Quarter 2004,pp.9-10.

[9] D.Vijendra Babu , P.Subramanian, C.Karthikeyan, "Performance Analysis of Block Matching Algorithms for Highly Scalable Video Compression" 1-4244-0731-2006 IEEE.pp.179-181

[10] Muhammed Z. Coban, Russell M. Mersereau, " Computationally Efficient Exhaustive Search Algorithm for Rate-constrained Motion Estimation" ,0-8186-8183-7/1997 IEEE pp.101-102

[11] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," Proc. NTC81, pp. C9.6.1-9.6.5, New Orleans, LA. Nov. 1981.

[12] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation,"IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313317, June 1996.

[13] M. F. So and A. Wu, "Four-step genetic search for block motion estimation," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process.(ICASSP '98), vol. 3. May 1998, pp. 1393–1396.

[14] Y.Nie, and Kai-Kuang Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, vol 11, no. 12, pp. 1442-1448, December 2002.