ISSN 2229-5518

A Novel QDS Search Algorithm for Block-Matching Motion Estimation in H.264 Video CODEC

Darshna D.Jagiwala, Prof. Mrs. S. N. Shah

Abstract— H.264/MPEG4 AVC video compression standard covers all common video applications ranging from mobile services and videoconferencing to IPTV, HDTV, and HD video storage. Motion estimation plays a significant function in digital video compression. However, its inherent computational complexity poses great challenge for real-time codec implementation. In this paper, a new search algorithm Quadratic Diamond Search (QDS) is proposed for reduction of computational complexity for motion estimation for H.264 codec. Experimental results demonstrate that the proposed method can allocate computation more accurately than previous methods. It will be shown that the proposed QDS algorithm achieves close performance but requires less computation by up to 44% on average than DS. Also the comparison of QDS with other fundamental block matching algorithms (BMA) which are Exhaustive Search (ES), Three Step Search (TSS), and Four Step Search (4SS) is given in this paper.

Index Terms— H.264 video CODEC, QDS algorithm, Fast Block Matching algorithms, Motion estimation, PSNR, Computations

1 INTRODUCTION

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)[1]. Motion estimation is the most time consuming part in the H.264 video coding standard. However, this part is crucial where it determines the quality and speed of an encoding process. Motion Estimation (ME) is a part of 'inter coding' technique. Inter coding refers to a mechanism of finding 'co-relation' between two frames (still images), which are not far away from each other as far as the order of occurrence is concerned, one called the reference frame and the other called current frame, and then encoding the information which is a function of this 'co-relation' instead of the frame itself.

A motion estimation unit at the video encoder in general includes the following three steps. In the first step, motion estimation is used to estimate the motion between blocks in the reference frames and a block in the current frame. The second step creates the displaced motion compensated frame and is referred to as motion compensation. The final step obtains the residual frame as the difference between the current frame and the displaced motion compensated frame. A similar procedure is performed at the decoder. The decoder first decodes the residual frame. Then it obtains the motion compensated frame using the motion vectors and the reference frames. Finally the difference frame is added to the motion compensated frame to obtain the reconstructed current frame. This reconstructed frame can be used as the reference frame for the next frames. Fig.1 shows the generalized block diagram of such hybrid video encoder [2].

In this paper, a new fast BMA is proposed, called the Quadratic Diamond Search (QDS) algorithm. The QDS algorithm will be explained in Section 2. In Section 3, the simulation results are given out and QDS algorithm is compared with other fast BMAs, and the conclusion is drawn in Section 4.

- *Darshna D.Jagiwala is currently pursuing masters degree program in EC engineering in SVNIT, Surat, Gujarat (INDIA), PH-09228282882. Email:darshna12@gmail.com*
- *Prof. Mrs. S. N. Shah is Assistant Professor in EC engineering in SVNIT, Gujarat(INDIA),ECED, SVNIT,Surat,Gujarat(INDIA),PH-09925458497 Email:snshah@eced.svnit.ac.in*

2 PROPOSED QUADRATIC DIAMOND SEARCH (QDS) MOTION ESTIMATION ALGORITHM FOR H.264 CODEC

2.1 Preliminaries

Block-matching motion estimation is performed between the current frame and a previously processed frame of a video sequence. The current frame is divided into non overlapped square blocks of pixels with size N X N and each block has a corresponding search area in the previously processed frame which has the size (2W + N +

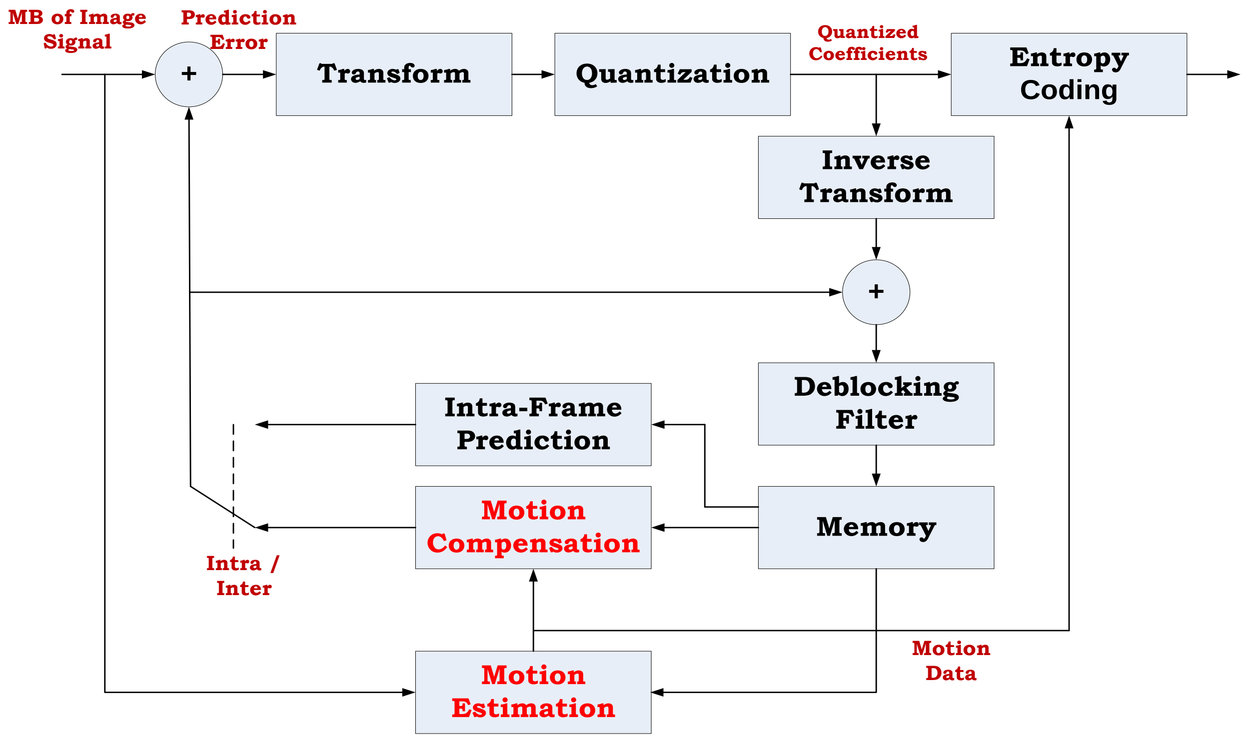

Fig.1 Generalized Block Diagram of a Hybrid Video Encoder

1) X (2W + N + 1), where W is the maximum displacement in pixels along both horizontal and vertical directions. Then, for each current block, we look for the block in the search area that best matches the current block at which the MAD has the minimum value, i.e., minimum error [3].The matching criterion is based on the mean absolute difference (MAD), which can be expressed as (1)

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

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

The PSNR block computes the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the value of PSNR, the better the quality of the compressed or reconstructed image is considered.

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

PSNR given by (2) characterizes the motion compensated image that is created by using motion vectors and macro blocks from the reference frame.

2.2 Fast Search Algorithms

By exhaustively testing all the candidate blocks within the search window, full search (FS) [4] algorithm gives the global optimum solution (i.e., the minimum matching error point over the search window) to the motion estimation, while a substantial amount of computational load is demanded. To overcome this drawback, many BMA's have been developed, for example, 2-D logarithmic search (LOGS) [5], three-step search (TSS) [6], conjugate direction search (CDS) [7], cross search (CS) [8], new three-step search (NTSS) [9], four-step search (4SS) [10], etc. These fast BMA's exploit different search patterns and search strategies for finding the optimum motion vector with drastically reduced number of search points as compared with the FS algorithm.

The DS [11] algorithm employs two search patterns. The first pattern, called Large Diamond Search Pattern (LDSP) shown in Fig. 2(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) shown in Fig. 2(b). In the searching procedure of the DS algorithm, LDSP is repeatedly used until the Minimum Block Distortion (MBD) occurs at the center point.

ISSN 2229-5518

There are still many drawbacks about DS algorithm. First, when the search pattern (LDSP or SDSP) is near to or at the search window boundary, the checking points outside the search window are truncated. Second, DS algorithm doesn't restrict the number of search steps essentially. To solve some drawbacks of DS, QDS algorithm is proposed in this paper.

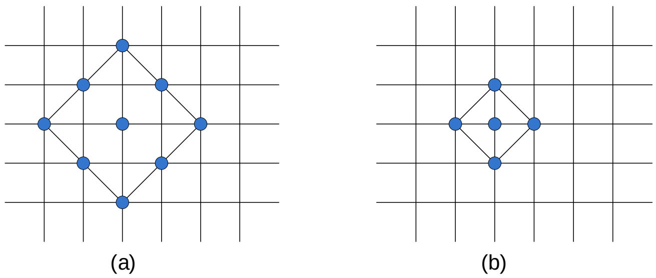

Fig.2 (a) Large Diamond Search Pattern (LDSP), (b) Small Diamond Search Pattern (SDSP)

2.3 The Proposed QDS Algorithm

In this algorithm, initially the search area is divided into four quadrants and the algorithm checks three locations X, Y and Z as shown in Figure 3. X is at the origin and Y and Z are Step size(S) = 4 locations away from X in orthogonal directions. Depending on certain weight distribution amongst the three the second phase selects few additional points (Fig 4). In particular, as shown in Fig. 4, the number of checking points for the quadrant search is five on average in the first step and four on average in the subsequent steps (excluding the location which is checked in the previous step). The rules for determining a search quadrant for second phase are as shown in Table 1:

TABLE 1
RULE FOR DETERMINING A SEARCH QUADRANT

Condition	Selected Quadrant
MAD (X) ≥ MAD (Y) & MAD (X) ≥ MAD (Z)	I
MAD (X) ≥ MAD (Y) & MAD (X) < MAD (Z)	II
MAD (X) < MAD (Y) & MAD (X) < MAD (Z)	III
MAD (X) < MAD (Y) & MAD (X) ≥ MAD (Z)	IV

Once we have selected the points to check for in second phase, we find the location with the lowest weight and set it as the origin. Then the search pattern is changed to SDSP with new step size S = S/2. The procedure keeps on doing SDSP until S=1. At that point it finds the location with the least cost function and the macro block at that location is the best match. The calculated motion vector is then saved for transmission. An example process is illustrated in Fig 5. The main advantage of this algorithm over DS is that it saves on computations by directly jumping to that quadrant and using SDSP, whereas DS takes its time doing LDSP. Although this algorithm saves a lot on computation as compared to DS, but in reality the error surfaces are not strictly unimodal and hence the PSNR achieved is poor compared to DS.

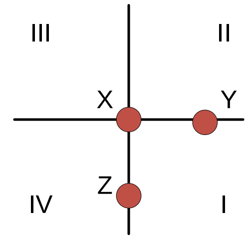

Fig.3 Search pattern in the first phase of each step of the QDS.

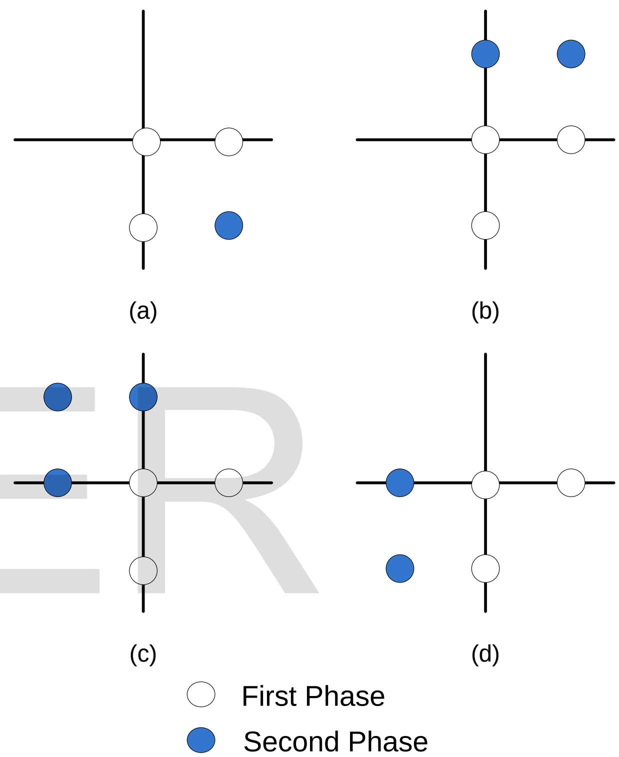

Fig. 4 Search patterns corresponding to each selected quadrant: (a) quadrant I, (b) quadrant II, (c) quadrant III, and (d) quadrant IV.

3 SIMULATION AND COMPARISON RESULTS

The original video is given to H.264 encoder with control of key parameters like GOP (Group of Picture), QP (Quantization Parameter), FPS (Frame per Second).Intra and inter prediction is applied on the video frames and residual energy is encoded. Intra and inter prediction is applied on the video frames and residual energy is encoded. The performance of the all above mentioned block matching algorithms are evaluated in terms of computation and average PSNR per frame of the reconstructed video sequence and it is computed for quality measurement. Here Computation is defined as the average number of the error function evaluations per MV generation. Then the residual energy is given to H.264 decoder and video is reconstructed and analysis is done on the based upon PSNR.

Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB [12], where higher is better.

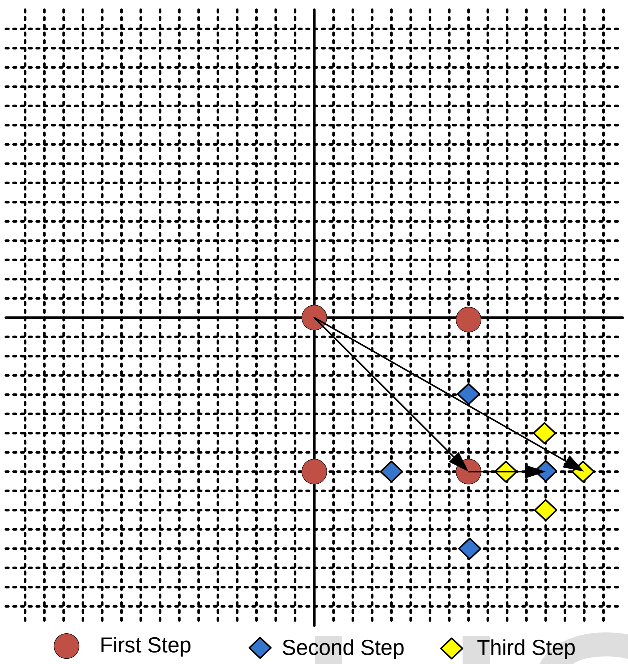

Fig. 5 The QDS procedure.
The motion vector is (4, 7) in this example.

To verify the usefulness and performance of the proposed method QDS, we compare it with other algorithms, namely, the FS, TSS, DS and 4SS on H.264 Video codec. 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 calculation. In the tests, IBBPBBPBBI GOP structure and QP (Quantization Parameter) = 10 are taken for encoding purpose in H.264 CODEC. Also All readings are taken with 4 X 4 Macro Block size and the value of search parameter is kept 7. Here MATLAB version R2010a software is used for programming and generating results.

The average of no. of computations for all motion algorithms and performance comparison of computations for all video sequences are included in Table 2 and Fig. 6, respectively. For software implementations, the computational burden of the full search algorithm is usually comparable to or greater than that of all remaining encoding steps combined. From the Table 2 and Fig. 6, it is proved that among these algorithms, highest computations are required for ES and lowest computations are required for QDS method.

PSNR is a measure of how good is the reconstructed image frame is when compared to the corresponding original frame. Table 3 represents the average value of PSNR for Fast Block Matching Algorithms. The PSNR comparison of the compensated images generated using the motion estimation algorithms is shown in Fig.7,8,9 for news, akiyo and shakycar respectively. Above result shows that ES provides highest value of PSNR as compared to other algorithms for all video sequences and TSS, 4SS and DS come pretty close to the PSNR results of ES. The PSNR value of QDS drops compared to all algorithms, but QDS takes up less number of search point computations amongst all. The results of ES, TSS, 4SS and DS are similar to that of [13].

TABLE 2

Average No. of Computations for All Video Sequences

Average No. of Computations for all Video Sequences					
Video Sequence	ES	TSS	4SS	DS	QDS
news	210.101	24.1205	16.6775	13.3754	5.9242
akiyo	210.101	24.1042	16.44	12.6768	5.9205
shakycar	216.333	24.5912	23.0731	19.8675	5.9515

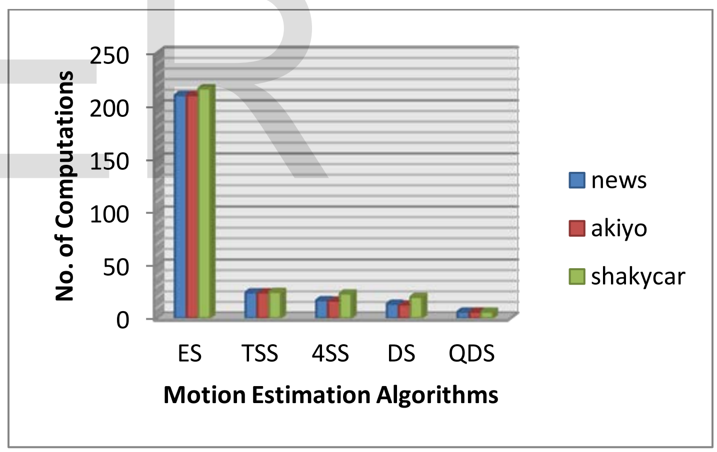

Fig.6 Performance Comparison Graph of Computations for All Video Sequences

TABLE 3
Average Value of PSNRs for All Video Sequences

Average No. of Computations for all Video Sequences					
Video Sequence	ES	TSS	4SS	DS	QDS
news	210.101	24.1205	16.6775	13.3754	5.9242
akiyo	210.101	24.1042	16.44	12.6768	5.9205
shakycar	216.333	24.5912	23.0731	19.8675	5.9515

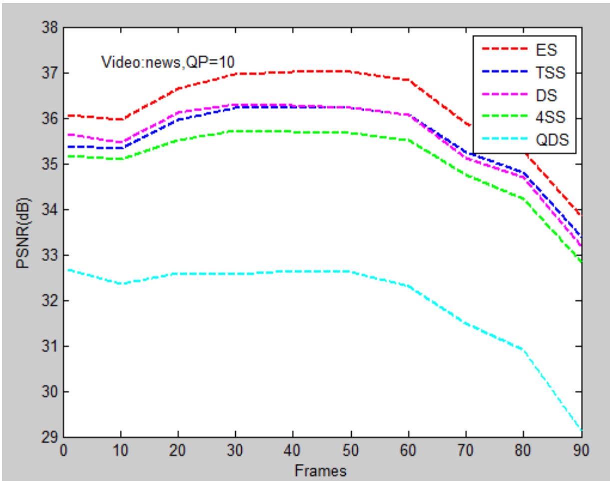

Fig.7 PSNR Performance of Fast Block Matching Algorithms for news Sequence using H.264 encoder

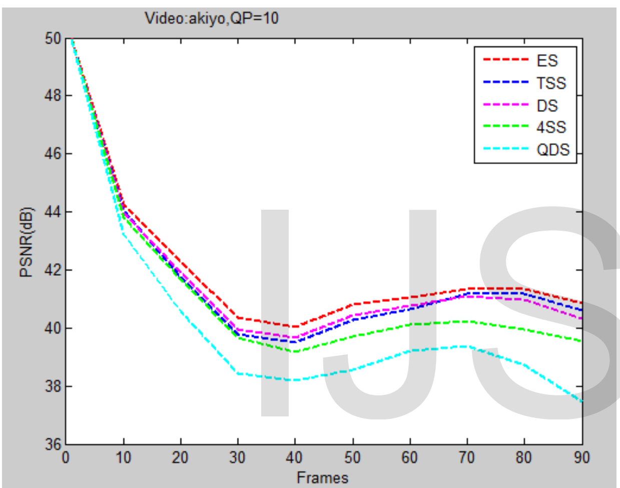

Fig.8 PSNR Performance of Fast Block Matching Algorithms for akiyo Sequence using H.264 encoder

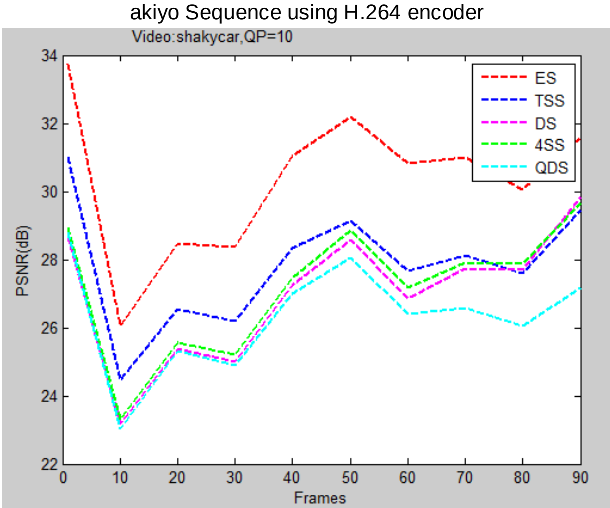

Fig.9 PSNR Performance of Fast Block Matching Algorithms for shakycar Sequence using H.264 encoder

CONCLUSION

In this paper, we have proposed a QDS algorithm for fast block matching motion estimation on H.264 CODEC. Full search Motion Estimation algorithm is not fit for real-time applications because of its unacceptable computational cost. As a consequence, the computation of H.264 video coding is greatly reduced with proposed algorithm. Experimental results based on simulations demonstrate that the QDS can provide good performance comparable with the ES, TSS, DS and 4SS in terms of computations.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their comments and suggestions.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, pp. 560-576, July 2003.

[2] 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,pp.8-10, 2004.

[3] S. Immanuel Alex Pandian, "A Study on Block Matching Algorithms for Motion Estimation", International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 1, pp.34-36, Jan 2011.

[4] Muhammed Z. Coban, Russell M. Mersereau," Computationally Efficient Exhaustive Search Algorithm for Rate-constrained Motion Estimation", IEEE pp.101-102, July-1997.

[5] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., vol. COMM-29, pp. 1799–1808, Dec. 1981.

[6] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in Proc. Nat. Telecommun. Conf., New Orleans, LA, pp. G5.3.1–5.3.5.,Nov. 29–Dec. 3 1981,

[7] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," IEEE Trans. Commun., vol. COMM-33, pp. 888–896, Aug. 1985.

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

[9] 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. 438–442, Aug.

ISSN 2229-5518

1994.

[10] 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. 313–317, June 1996.

[11] Shan Zhu and Kai-Kuang Ma," A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," IEEE Trans. On Image Processing, vol. 9, No. 2, pp.287-290,February 2000.

[12] T. Oelbaum, V. Baroncini, T. K. Tan, and C. Fenimore, "Subjective Quality Assessment of the Emerging AVC/H.264 Video Coding Standard", International Broadcasting Conference (IBC), Sept.2004.

[13] Aroh Barjatya, "Block Matching Algorithms For Motion Estimation", IEEE. DIP 6620 Spring Final Project Paper, pp.6, 2004.