International Journal of Engineering & Technology, 7 (2.12) (2018) 394-398

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper



Literature survey on motion estimation techniques

S. Mahaboob Basha [1 *], M. Kannan [2]

[1] Assistant Professor, Department of Electronics and Communication Engineering, Velammal Engineering College Chennai, India.
[2] Associate Professor, Department of Electronics Engineering, MIT, Anna University Chennai, India.
**Corresponding author E-mail: phoenix_basha@yahoo.com*

Abstract

Motion Estimation (ME) is one of the most intensive computational operations in video compression techniques. Video compression algorithm utilizes numerous standards such as MPEG1, MPEG4 AND H.261, H.264. Compression performance can be increased drastically by efficient motion estimation techniques by which energy is reduced within the residual frames involved in motion compensation. In this paper literature survey of motion estimation especially considering block matching ME (Motion Estimation). In this paper, comparison is made between the already existing block matching algorithms and their limitations in motion estimation along with their applications. Many algorithms including Three Step Search (TSS), Improved Three Step Search (ITSS), New Three Step Search (NTSS), Four Step Search (4SS), Diamond Search (DS), Flexible Triangle Search(FTS), Full Search(FS), Modified Full Search(MDF) are compared and their per-formance measures are discussed in this paper.

Keywords: *Block Matching Motion Estimation; DS; TSS; ITSS;FTSS; FSS; DS; FTS.*

1. Introduction

Block Matching motion estimation is employed for its efficiency and used for motion-compensated coding of image sequence. Shift Register Arrays (SRA) and a systolic array processor are included in the proposed model which can reduce the pin counts while performing parallel processing, also operates in real time video conference applications and it is a modular and simple design.

Motion can be estimated by a block-by-block basis by employing the popular technique of Block Matching Algorithm (BMA). Block of pixels in the current frame is compared with the corresponding block of pixels in the reference frame within the search area, their MAD (Mean Absolute Difference) values are calculated and from the matched best block having least MAD the Motion Vector (MV) can be calculated [1].

Researchers have found several algorithms for motion estimation. Superior to the previous Three Step Search (TSS) and New Three Step Search (NTSS) algorithms, an Improved Three Step Search (ITSS) motion estimation algorithm with in-built flexible and dynamic VLSI architecture is proposed and is used mainly for low bit rate video coding especially for H.261 standard.

ITSS algorithm utilizes FPGA addressing circuit and VLSI tree processor and is implemented with minimum number of gates. Due to the ITSS architecture flexibility, other three-step search algorithms (TSS, NTSS) also could be implemented by using ITSS motion estimation algorithm [2].

In comparison with the standard Three Step Search (TSS) algorithm, (FTSS) Fast Three Step Search reduces the total number of checking points involved in motion estimation. This architecture uses intelligent memory configuration and data arrangement and involves only three processing elements. In FTSS, the previous motion vector's directional information is used and current motion vector is estimated. Total number of checking points is reduced by using the directional information in FTSS while ensuring better performance than TSS [3].

Full Search and Fast Search algorithms can also be used in motion estimation. BMA based full search is performed between the reference and current frame and block with most similarity (SSD value is minimum) is selected and the difference of two blocks and their motion vector is transmitted. For reducing the computational complexity, fast search algorithms are used instead of full search algorithms.

These fast search algorithms reduces the number of check points by using special pattern for motion estimation like diamond pattern and Hexagonal pattern. In DS (Diamond Search) method, the search points can be further reduced based on the SAD (Sum of Absolute Difference) values calculated in the previous step and based on the values obtained either LDSP(Large Diamond Search Pattern) or SDSP(Small Diamond Search Pattern) can be used. This technique is called as MVFAST (Motion Vector Field Adaptive Search) technique.

Another fast search algorithm will eliminate the calculations of ME's for some types of blocks by differentiating the SKIP mode from the other types of blocks and gives highest priority is given to SKIP mode. SKIP mode has the lowest complexity and motion search is not required for this mode [4].

FTS(Flexible Triangle Search) algorithm was proposed which finds the best block between the current and reference frames by using a search triangle whose size and direction can be changed through various operations. Finite set of triangles with different orientations are used to perform fine searches. The operations like translation, reflection, expansion , contraction are used for changing the direction, location and size of the searching triangle. This algorithm achieves higher frame rates and is more efficient in terms of power consumption [5].

Copyright © 2018 S. Mahaboob Basha, M. Kannan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

2. Discussion about review papers

The function of video compression system utilising various Block based matching and Full search block estimation techniques are critically briefed in this section.
In Fig 1. the motion estimation technique and determination of motion vector are shown which are used in block matching analysis.

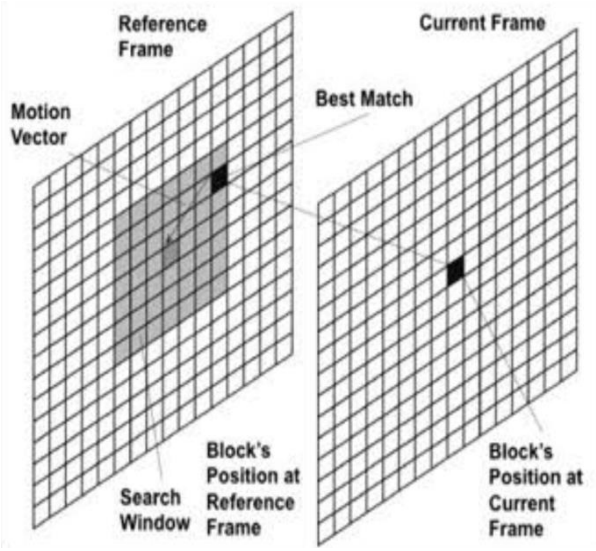


Fig. 1: Motion Estimation Technique and Motion Vector Determination.

Paper [6] (P.Muralidhar and C.B Ramarao) states that MFSBMA (Modified Full Search Block Estimation Algorithm) is preferred over FSBMA(Full Search Block Estimation Algorithm) where MFSBMA maintains the same PSNR(Peak Signal To Noise Ratio) as that of FSBMA while doing the SAD(Sum of Absolute Difference) calculations for the current foreground micro block but skipping the SAD calculations for the current background micro block. Background elimination algorithm is implemented which threshold each BF (Background Frame) pixel value, if the pixel value of BF is less than the threshold, it means the CF(Current Frame) pixel is a background pixel and its value is replaced by 0 else if the value of BF pixel is larger compared to the threshold then CF pixel is a foreground pixel and motion vectors are calculated only for the foreground pixels thereby reducing the SAD calculations.
Paper [7] (V. S. K. Reddy, S. Sengupta, and Y. M. Latha) implements FSBM by employing parallel bank architecture which provides maximum throughput and full hardware utilization with minimum latency. The proposed VLSI Architecture is divided into four parts namely address generator, on-chip buffer, computation core and off-chip memory. Nine parallel processors are used which is controlled by a Moore Model FSM (Finite State Machine.
The major module is the state machine module that has four states like Idle State, Block Loader State, SAD Generator State, comparator States and the last module controls all the other modules. This VLSI architecture is easily scalable and parallel implementations obtain higher speed.
Paper [8] (Viet L. Do and Kenneth Y.Yu) proposed an enhancement in the FSBM motion estimation architecture by using conservative approximation the power consumption could be reduced by a factor of 2 and eliminates unnecessary computations while maintaining optimal throughput. In this approach, before computing the exact distortion, a conservative estimate of the exact distortion value is calculated for each candidate micro block.
The exact distortion need not be calculated if the distortion's conservative estimate is larger when compared to the minimum exact distortion found so far. The power consumption is reduced without affecting the accuracy of the performance by employing conservative approximation method in FSBM motion estimation.
In Paper [9] , algorithm employed makes use of pixel averaging known as Global Elimination(GE) algorithm.GE employs two stages, in the first pre-processing stage the comparison of SSAD (Sub-Sampled SAD) with the minimum SAD takes place where the unnecessary SAD(Sum Of Absolute Difference) calculations can be skipped. In the second stage, at approximately and randomly matched candidate positions SAD will be calculated. Fig 2. And Fig 3. Shows the possible partitions of a 16x16 macro block and the possible sub partitions of a 8x8 macro block.
For computing SSAD values in the first pre-processing stage, Block-sizes and shapes that are used are fixed which is one of the limitations of Global Elimination algorithm. This complexity of GE algorithm can be reduced drastically by using blocks that can adaptively change their sizes depending on the features of the macro block. 50% high throughput and 60% less computations are achieved in adaptive Global Elimination Algorithm.
Since SSAD is used as a matching measure between the current macro block and reference macro block (MB) rather than complete SAD, complexity reduction is easily achieved.
Top 'N' macro block candidates are selected for the second stage which implements the FS (Full Search) technique on the entire 'N' block candidates and determines the best motion vector. In adaptive GE, the right amount of partitioning is found by calculating the pixel variance of each MB. Smaller number of partitions are used for lower variance MB's while Higher variance MB's will have larger number of partitions. Adaptive GE resulted in drastic reduction of SAD Calculations.
Paper [10] introduced a novel search algorithm for full search known as Quarter pel full search block motion estimation. This design calculates all the 41 motion vectors in parallel that are required by the various sub blocks. New features including quarter sample accuracy, multiple reference frames for estimation and variable block sizes are discussed in this paper. This proposed architecture can process nearly 5 reference frames in parallel.
In this system, for search window a single port block RAMs are used whose size is 92x92(Quarter pel) pixels and motion vector value lies between -3.75 to +4.00 for 16X16 size blocks. The architecture also includes Shift registers, eight processing units, Address Generator (AG) and comparing units. For real time processing of CIF (Common Intermediate Format) Images, this pipelined and parallel design is efficient.
There are 41 Parallel In Serial Out (PISO) Shift Registers and each PISO computes the SAD values for one Particular block size/type and finally fetched to the comparing unit to get the best motion vector as output.

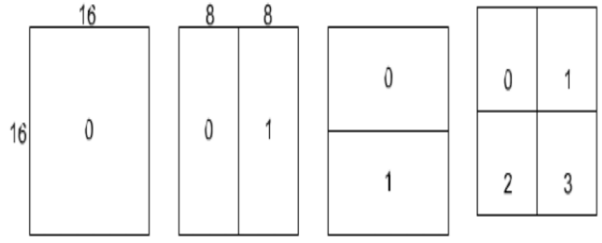


Fig. 2: 16x16, 8x16, 16x8, 8x8 Macro Block Partitions.

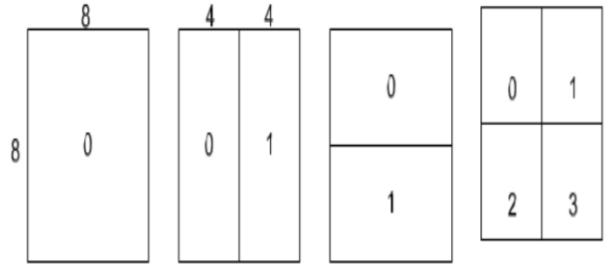


Fig. 3: 8x8, 4x8, 8x4, 4x4 Macro Block Sub Partitions.

Paper [11] (Ranjan Maity) portrays that by exploiting the data redundancies like Spatial Redundancy(known as Intra-Frame Redundancy) due to highly correlated pixel values in a video sequence and Temporal Redundancy(known as Inter-Frame Redundancy) which exists due to the correlation of pixel values between the successive frames, efficient compression of video sequences can be achieved.

DCT (Discrete Cosine Transform), DWT (Discrete Wavelet Transform) and HWT(Haar Wavelet Transform) are techniques for removing spatial redundancy and Motion Estimation technique is employed for temporal redundancy removal.

HEXBS (Hexagon Based Search) is employed in this paper for finding the motion vector and a threshold value is chosen which is not fixed but varies dynamically. In HEXBS, to search patterns such as LHSP (Large Hexagonal Search Pattern) and SHSP (Small Hexagonal Search Pattern) are involved.

Paper [12] emphasizes the necessity and importance of the Motion estimation with variable block sizes to get high encoding quality. Variable Block Size (VBS) MCP (Motion Compensated Prediction) gives efficient performance in terms of rate distortion when compared to the conventional Fixed Block Size MCP.

16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4 are considered to be the different types of VBS which leads to a total combination of 41 possible block sizes. Based on the search area's memory size there are many overlapping macro block candidates in calculating the 41 motion vectors for all possible combinations of 16x16 macro block. Hence different scan patterns like meander scan, raster scan, spiral scan or z scan patterns can be used in reading macro block pixels from different memory before SAD computation.

In VBSME (Variable Block Size Motion Estimation) raster scan(both horizontal and vertical) patterns are traditionally used in which 1,4 or 16 pixels are read simultaneously and given to processing elements to generate SAD combinations. The proposed architecture uses z scan pattern over raster pattern.

In this paper, 16-pixel z scan pattern is used to access current MB pixels and 17 Candidate MB pixels to compute all 41 possible combinations for a 16x16 block in minimum number of clock cycles(here 16 clock cycles are used). The process is repeated for17 times hence in 272 clock cycles all Candidate block combinations are available , best match is found and Motion Vector is computed.

Paper [13] (Swee Yeow Yap and John V. McCanny,Fellow) proposed a new (1-D) One Dimensional very large scale implementation for full search termed as FSVBSME(Full Search Variable Block Size Motion Estimation). In this 1-D architecture, for subblocks 41 Motion Vectors are processed in same number of clock cycles whereas only one MV is processed in conventional 1-D architecture.

Multiplexers, Latches with less complex computational hardware is required in this system. CMD(Current Macro block Data) is arranged in a raster scan pattern and SRD(Search range Data) is also arranged in a dual raster scan pattern and proposed approach calculates 16 SAD values for 4x4 size blocks which can be reused while computing larger block sizes SAD values.

The key aspect of this paper is that, within each Processing Element(PE) , shuffling and combining partial SAD values mechanisms are incorporated. As a result, SAD for larger blocks can be calculated from the results that are derived from the 4x4 blocks without performing the entire computation from the scratch.

In paper [14] (S. Rukmani Devi, P. Rangarajan, and J. Raja Paul Perinbam), EDS(Enhanced Diamond Search) is proposed for fast block motion estimation. LCDP (Large Cross Diamond Pattern) contains 5 search points with ±2 pixel distances whereas SCDP(Small Cross Diamond Pattern) contains 4 search points with ±1 pixel distance.

This algorithm includes only few search points and PSNR value is also significantly increased. EDS is implemented in FPGA device with fixed block size motion estimation.

Paper [15](K.Leela Bhavani and R.Trinadh) presents a novel Block Matching technique called as ARPS(Adaptive Rood PAttern Search). Two search stages are implemented in this technique namely Initial Search and Refined Local Search. ARD pattern is used for initial search and this pattern is symmetrical and spreads in horizontal and vertical directions. There are six search points in the initial search stage and then 5 points for the refinement process including the predicted Motion Vector from the initial searchif that MV matches with the target MV.

The ARPS search method is superior over Diamond Search (DS). The key aspects of the design used are the adaptability of the rood arm and with the help of the predicted motion vector, if the current Macro Block belongs to same object as that of its neighboring block , then the search can be terminated with the initial search stage itself and in the alternate case when MB does not belong to the same object as of its neighbor, then pattern searches points in the vertical directions and horizontal directions to find the motion vector.

Paper [16] (Harsha Prakash Redkar, Sonia Kuwelkar), implements two methods for SAD calculations namely bit parallel method and bit series method. Bit parallel method gives faster results for SAD and higher throughput with the design being implemented on FPGA.

In bit series method, 16 pixels are processed simultaneously hence 4x4 blocks SAD can be calculated which can be reused to compute larger block SAD. Online Arithmetic (OLA) is used for computations, where the results and digit-by-digit operands are overlapped. The OLA method uses radix-2 signed digit representation with the 1st bit weighted positively and the 2nd bit is weighted negatively. SAD value is calculated by finding the difference between these digits and one more step is required to convert to signed digit form.

In bit parallel method, AVC (Absolute Value Calculator) and Row-SAD unit are used. The advantage of this method is no need of signed digit conversion and only pixels with greater value are determined to find the absolute difference. 16 AVC units are used in parallel to give 16 absolute values. Hence less number of clock cycles are required for bit parallel method compared to bit series method.

In paper [17] (Chen-Kuo Chiang, Wei-Hau Pan, Chiuan Hwang , Shin-Shan Zhuang and Shang-Hong Lai) , fast algorithms are proposed based on statistical learning for computational efforts reduction required in 3 major components like multi-reference Motion Estimation, inter-mode decision and intra-mode prediction .

Representative features are first extracted and an offline pre-classification approach determines the best results from the features that are extracted, thus energy computation is reduced by a significant amount and this method is faster compared to the conventional FS method.

Best Inter SAD, MV Difference and MV magnitude, Best Intra SAD and Gradient Magnitude are the features extracted by statistical learning.

The possible sub-blocks of 16x16 and 8x8 sub macro block are preclassified by using SVM(Support Vector Machine). In inter-mode decision depending on representative features and pre-classified sub blocks an algorithm is proposed to perform SAD calculations.

In MR (Multiple Reference) ME based on the data from the look-up table collected by SVM and features , the number of reference frames to be compared is decided. In Intra-mode prediction exhaustive search is applied to determine the optimal intra-modes. In case of Luma 4x4, the possible modes range from 0 to 8 and in Chroma, the possible modes range from 0 to 3.

Paper [18] (Hyo-Sung Kim, Jong-Hyeok Lee, Chang-Ki Kim, and Byung-Gyu Kim), proposed a novel ZME(Zoomed Motion Estimation) and Motion Compensation Technique where technique of local area scaling is used. A ZV(Zoom Vector) is introduced by which the reference block size either decreases or increases and in order to perform block matching, this resized block is reformed to the size of the current block.

In ZV, for eliminating unnecessary search points, a 3-D Diamond search pattern is used. Higher bit-rate savings can be achieved with efficient peak signal-to-noise ratio.

Paper [19] (Ling Li, Shaoli Liu, Yunji Chen, Tianshi Chen and Tao Luo), discusses about motion estimation that does not consider Integer-Pel Search. Three main steps for ME includes spatial-temporal prediction, fractional pel search, Integer pel search is crucial in video encoding process.

This paper introduced a novel approach of encoding video sequences without integer-pel search. This approach is optimal over conventional approach including the integer-pel search only to fina near MV(Motion Vector) stand-alone.

In paper [20] (Shiaw-Yu Jou, Shan-Jung Chang, and Tian-Sheuan Chang), highly efficient ME design is proposed with joint architecture with an algorithm. Predictive IME(Integer ME) algorithm is

employed to reduce the complexity which by using statistical analysis select only the most probable search directions thereby drastically reducing the number of search points. Also a size-dependent FME (Fractional ME) with a PU (Processing Unit) is employed to the interpolation filtering.

For an early MV prediction approach , the IME and FME computations through interlaced scheduling are cascaded. Cache controller is simplified by a double Z-scan indexed addressing thus reducing the Bandwidth. Cost efficient and a high-throughput design involving reduced gate-count and on-chip memory size is presented in this paper.

In paper [21] (Zhaoqing Pan, Yun Zhang, Member and Sam Kwong, Fellow) , to reduce the computational complexity of MVC(Motion Video Coding), an efficient Disparity Estimation(DE) and Motion Estimation(ME) algorithms are proposed. With respect to the rate distortion and coded block pattern, early direct mode decision algorithm is implemented and then based on the characteristics of initial search point in DE/ME technique, if search point is center-biased , the search window for ME/DE process is decreased or early ME/DE termination technique is employed. Finally, best match point for DE/ME process is predicted by considering two block matching methods.

Paper [22] (Nan Cen, Zhangyu Guan and Tommaso Melodia), states that multi view video streaming applications which are wireless in nature like 360degrees videos, a novel multi view encoding/decoding is used. On the basis of a new Motion Compensation algorithm, at the decoder side SI (Side-Information) method is designed for multi view video joint reconstruction.

The received measurements along with resampled measurements generated from SI is fused by a technique in order to perform the final recovery. A blind video quality estimation technique is also derived from this joint reconstruction technique to adapt online the encoding rate of the videos in multiview video streaming. The reconstruction quality can be accurately estimated by the proposed blind estimation algorithm.

Paper [23] (Xin Jin, Satoshi Goto and Life Fellow), emphasizes the fact of low power consumption of video encoding considering real-time videos to improve quality of service. Inefficiency exists in video encoding since the retrieval of compressed information before encoding is not possible and non-deterministic nature of future video content is also a reason for inefficiency.

For video encoding applications, a workload estimation scheme has been proposed. The estimation technique is Hilbert transform-based workload estimation model which predicts the overall variations in encoding workload and those fluctuations are resolved.

Paper [24] (Zhan Qu, Jinqiao Wang, MinXu and Hanqing Lu), states that temporal-spatial requirements which preserves temporal coherence, spatial geometry are dealt by the existing video retargeting techniques. In this paper, graph model is employed to propose a novel context-aware solution.

Firstly, to preserve the temporal motion and spatial structure a grid-based warping framework is used at each grid cell. Then a graph-based motion partition algorithm is employed to estimate motions of various regions, which by simultaneously estimating motions of regions also provides the contextual relationship between grid cell. For the same object's grid cells undergoing uniform temporal- spatial transformation 2 novel constraints are encoded for spatial-temporal preservation.

Finally, quadratic programming problem is formulated the objective function. The proposed method improves the speed because grid-cell wise ME can be calculated for every few frames and satisfactory spatial-temporal coherence is achieved avoiding all the artifacts influence.

Paper [25] (Huynh Van Luong, Lars Lau Rakêt and Søren Forchhammer), to improve the side information that depends on the optical, noise residual frames considering partial information, a motion re-estimation technique has been deviced called as noise residual motion re-estimation technique for improving noise modelling. From previously decode frames in order to estimate a current residue and for correlation between estimated side frames Residual Motion Compensation with motion updating is used.

The proposed MORE(Motion Re-Construction and Re-Estimation) techniques uses noise learning and side information. This MORE codec technique provides efficient PSNR improvement over other methods.

Comparison between various proposed algorithms

The performance of various reviewed algorithms and the comparison results for the tested Akiyo, Stock halm and Foreman video files are summarized in Table 1.

Table 1: Performance Comparison between Various Algorithms Discussed in [4], [3], [15], [14], [6]

Year	Algorithm	Video File Considered	PSNR(db)	Application	Drawback
2008	FS2 (SKIP mode Considered)	Foreman	30.01	Operates with larger sized blocks and has SKIP mode to reduce SAD calculations.	Not efficient for larger values of QP(Quantization Parameter).
2008	DS (LDSP and SDSP considered)	Foreman	31.03	Faster than FS2(SKIP MODE) for smaller sized blocks.	Not efficient for smaller values of QP(Quantization Parameter).
2012	ARPS	Stock halm	30.54	More efficient than DS and in terms of search points.	Rood arm can trace only in horizontal and vertical directions.
2012	TSS	Akiyo	40.543	Less Search points and efficient for centre biased search.	PSNR values obtained are not efficient and lack of accuracy.
2012	FTSS	Akiyo	41.644	Reduction in computational complexity and power dissipation and preferred over TSS.	Suitable only for low bit rate and low power video applications
2012	FS	Akiyo	42.744	High efficiency in PSNR values and accurate method.	Complexity in terms of search points and large number of computations.
2013	EDS	Foreman	33.615	Increase in PSNR value compared to conventional DS algorithm.	This method is considered only for fixed block size ME and not efficient for variable block size ME.
2015	FSBMA (FSBMA Vs MFSBMA)	Akiyo	38.04	Number of SAD calculations reduced by 50% compared to FS without affecting Performance measures.	Hardware utilization is high compared to full search method.
2015	MFSBMA (FSBMA Vs MFSBMA)	Akiyo	40.63	Efficient in terms of finding the best match by calculating SAD for all the reference blocks with respect to current block.	High computational complexity and increased calculations.

3. Conclusion

From reviewing and comparing various algorithms for motion estimation FS and MFSBMA algorithms are found to be efficient with the latter considering half the search points required for the FS method. Other algorithms like FTSS, ARPS also provide efficient results with acceptable image quality degradation and drastically reduce the number of search points considered for SAD calculations. EDS algorithm is giving good results when compared to the existing DS search algorithms. By minimizing the search points block based motion estimation algorithms speed and throughput can be improved. Achieving high accuracy with reduced search points and involving variable block sizes is the major challenging factor that has to be accomplished by various motion estimation algorithms for Video Compression Technology.

References

[1] Chaur-Heh Hsieh and Ting –Pang Lin, VLSI Architecture for Block-Matching Motion Estimation Algorithm, IEEE Transactions on Circuits and Systems for Video Technology, Vol.2, NO.2, June 1992.

[2] Xu, D. and Bentley, J. P. (2001), VLSI-based parallel architecture for block-matching motion estimation in low bit-rate video coding, The 8th IEEE international conference on electronics, circuits and systems, Malta, September 2 - 5, 2001 in Conference proceedings, vol. I-III. IEEE, pp.217-220.

[3] Namrata Verma, Tejeshwari Sahu, Pallavi Sahu, Efficient Motion Estimation by Fast Three Step Search Algorithms, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 1, Issue 5, November 2012.

[4] A.Ahmadi , M.M.Azadfar, Implementation of Fast Motion Estimation Algorithms and Comparison with Full Search Method in H.264, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.3, March 2008.

[5] M. Rehan, M. Watheq El-Kharashi, P. Agathoklis, and F. Gebali, An FPGA Implementation of the Flexible Triangle Search Algorithm for Block Based Motion Estimation, ISCAS conference 2006.

[6] P.Muralidhar, C.B. Ramarao, High Performance Architecture for Full Search Block Matching Algorithm, IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 5, Issue 1, Ver. II (Jan - Feb. 2015), PP 41-49.

[7] V. S. K. Reddy, S. Sengupta, and Y. M. Latha, New VLSI Architecture for Motion Estimation Algorithm, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:1, No:12, 2007.

[8] Viet L. Do and Kenneth Y. Yun, A Low-Power VLSI Architecture for Full-Search Block-Matching Motion Estimation, IEEE Transactions On Circuits and Systems for Video Technology, Vol. 8, No. 4, August 1998.

[9] P.Muralidhar and C.RamaRao, Efficient architecture for global elimination algorithm for H.264 motion estimation, Indian Academy of Sciences, Vol.41, No.1, January 2016, pp. 47-54.

[10] C.A.Rahman, Wael Badawy, A quarter pel full search block motion estimation architecture for H.264/AVC, ICME conference , August 2005.

[11] Ranjan Maity, Synthesis of Low Power High Performance VLSI Architecture for Video Codec Chip, Synopsis seminar report submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology.

[12] Nehal N. Shah, Harikrishna Singapuri, and Upena D. Dalal, Hardware efficient Architecture with Variable Block Size for Motion Estimation, Hindawi Publishing Corporation Journal of Electrical and Computer Engineering Volume 2016, Article ID 5091519, 11 pages.

[13] Swee Yeow Yap and John V. McCanny,Fellow, A VLSI Architecture for Variable Block Size Video Motion Estimation, IEEE Transactions on Circuits and Sysytems—II: Express Briefs, Vol. 51, No. 7, July 2004.

[14] S. Rukmani Devi, P. Rangarajan, and J. Raja Paul Perinbam, VLSI Implementation of High Performance Optimized Architecture for Video Coding Standards, Acta Polytechnica Hungarica, Vol. 10, No. 6, 2013.

[15] K.Leela Bhavani and R.Trinadh, Architecture for Adaptive Rood Pattern Search Algorithm for Motion Estimation, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 1 Issue 7, September – 2012.

[16] Harsha Prakash Redkar, Sonia Kuwelkar, Full Search Block Matching Algorithm Motion Estimation on FPGA, International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 4, Issue 4, April 2015.

[17] Chen-Kuo Chiang, Wei-Hau Pan, Chiuan Hwang , Shin-Shan Zhuang and Shang-Hong Lai, Fast H.264 Encoding based on Statistical Learning, IEEE Transactions on Circuits and Systems for Video technology, Vol. 21, No.9, September 2011.

[18] Hyo-Sung Kim, Jong-Hyeok Lee, Chang-Ki Kim, and Byung-Gyu Kim, Zoom Motion Estimation Using Block-Based Fast Local Area Scaling, IEEE Transactions on Circuits and Systems for Video technology, Vol. 22, No.9, September 2012.

[19] Ling Li, Shaoli Liu, Yunji Chen, Tianshi Chen and Tao Luo, Motion Estimation Without Integer-Pel Search, IEEE Transactions on Image Processing, Vol. 22, No.4, April 2013.

[20] Shiaw-Yu Jou, Shan-Jung Chang, and Tian-Sheuan Chang, Fast Motion Estimation Algorithm and Design for Real Time QFHD High Efficiency Video Coding, IEEE Transactions on Circuits and Systems for Video technology, Vol. 25, No.9, September 2015.

[21] Zhaoqing Pan, Yun Zhang, Member and Sam Kwong, Fellow, Efficient Motion and Disparity Estimation Optimization for Low Complexity Multiview Video Coding, IEEE Transactions on Broadcasting, Vol. 61, No.2 June 2015.

[22] Nan Cen, Zhangyu Guan and Tommaso Melodia, Interview Motion Compensated Joint Decoding for Compressively Sampled Multiview VideoStreams, IEEE Transactions on Multimedia, Vol. 19, No.6 June 2017.

[23] Xin Jin, Satoshi Goto and Life Fellow, Hilbert Transform-Based Workload Prediction and Dynamic Frequency Scaling for Power-Efficient Video Encoding, IEEE Transactions on Integrated Circuits and Systems for Video technology, Vol. 31, No.5, May 2012.

[24] Zhan Qu, Jinqiao Wang, MinXu and Hanqing Lu, Context-Aware Video Retargeting via Graph Model, IEEE Transactions on Multimedia, Vol. 15, No.7, November 2013.

[25] Huynh Van Luong, Lars Lau Rakêt and Søren Forchhammer, Re-estimation of Motion and Reconstruction for Distributed Video Coding, IEEE Transactions on Image Processing, Vol. 23, No.7, July 2014.