An Efficient Full Search Block Motion Estimation using Trace and Sum of Off- Diagonal based Match and FPGA Implementation

Manu T.M
K.L.E Institute of Technology,
Hubballi, Karnataka-580030

Linganagoud Kulkarni, Ph.D.
1B.V.B College of Engineering,
Hubballi-16

Basavaraj. S. Anami, Ph.D.
K.L.E Institute of Technology,
Hubballi, Karnataka-580030

ABSTRACT
The H.264/AVC offer many coding tools that give higher compression ratio compared to earlier standards. These tools increase the computational complexity of the motion estimation. In this paper, we have presented a computationally efficient method to speed up the search process by minimizing the computational load on general full search. The proposed method is having two phases, wherein the first phase involves search for the nearly matching block using the trace and sum of off-diagonal elements. Only upon match, the Sum of Absolute Differences (SAD)is calculated for best motion prediction in the second phase and it is only for limited number of blocks. The trace and off-diagonal sum matching requires O(2n) computations and $O(n^2)$ computations are required in case of full search. Hence, the proposed method reduces 75% to 80% of SAD computations of full search motion estimation. This leads to significant improvement in coding efficiency compared to the existing motion estimation techniques without much degradation in video quality. FPGA implementation result shows that, the design can work at the maximum clock frequency of 420 MHz with the power consumption of 34.86mW

Keywords
Video frames, Motion estimation, temporal prediction, Trace, SAD calculation, Off-diagonal sum.

1. INTRODUCTION
In video encoding, motion estimation (ME) is the most computation demanding part that affects the video quality, compression efficiency and speed [2]. For these reasons, many algorithms and architectures are proposed to improve above parameters. Most of the time tradeoff is made between quality and speed in implementation of ME algorithms because, ME is highly scene dependent. So, it is not advisable to rely upon a single technique to generate a good visual quality video for all kinds of scenes. Instead, it demands for combination of variety of architectures and algorithms with parameters, such as motion starting point, motion search patterns, and adaptive control to curb the search [3]. Thus, in video processing standards, the requirement of optimized methods for motion estimation is high. Hence, there is scope for development of efficient software algorithms and hardware architectures.

The rest of the paper is organized into six sections. The literature survey is given in section 2. In Section 3, the full search block matching motion estimation is described. In section 4, the proposed a trace match based methodology for full search motion estimation is presented. In section 5, hardware synthesis of the proposed methodology is given. The experimental results and discussion are given in Section 6. The conclusions are drawn in Section 7.

2. LITERATURE SURVEY
A considerable amount of research has been carried out to develop a fast block matching techniques that finds a suitable match by using few evaluations, as in case of three step search [3], novel four step search [4], the cross search [3],the diamond search technique [3], the hexagon based search technique [4], and block-based gradient descent search technique [5]. In all these techniques, the search is done only on some of the candidate blocks from the search area and chosen for a match from this subset of blocks only in some fixed point positions, because of which their result errors in motion prediction. Any error in motion prediction leads to incorrect motion vectors, high mean square error (MSE) of the motion compensated frame, which further degrades peak-signal- to- noise- ratio (PSNR).

To achieve a tradeoff between the computational complexity and degraded PSNR, the recent research works are found on reducing the computational complexity of fast search block motion estimation (FSBME), without causing much reduction in PSNR. A modified FSBME works on a data driven threshold that updates according to the picture variations, has given four times less computations compared to other fast search (FS) methodologies [14]. Another computation reduction technique, wherein the motion vector is determined based on the constant and reference thresholds [11]. A method, which lower bounds the matching criteria for subdivided blocks and reduces the number search positions is given in [15]. A new predictive search area approach is proposed in which the predicted search area is obtained from sub area of the neighboring blocks [12]. It is also possible to reduce further by employing a predicted region search. A method is found wherein a full search range is divided into overlapping regions ensuring the motion vectors close to the boundary of the two regions that are determined from either of the two regions [17].

In all these technique, search is mainly carried out based on the assumption that the matching error decreases monotonically, when approaching towards the global best points and error drops down to local minimum points. All these techniques are lacking regular data flow and the predictable terminations. These techniques suffer from large quality degradation, when the motion field is large and complex. Hence, to overcome these drawbacks, we have developed a methodology to minimize the computational overheads of FSBM without degrading the video quality.

A two phase method to minimize the computations in estimating the motion vectors is developed and implemented. Instead of exploiting the motion activity by calculating SAD between adjacent frames directly, we have calculated the motion activity by comparing the trace and sum of off-diagonal elements of the current block to the trace and sum off diagonal elements of the candidate block in the first phase. If these values are exactly equal or comparable according to the predefined threshold value, then the two blocks are considered matching otherwise they are unequal or mismatching. Upon trace match, the best matching block is found by calculating the SAD in the second phase. Thus, the numbers of SAD value to be computed is reduced significantly.

3. FULL SEARCH BLOCK MATCHING MOTION ESTIMATION

In H.264/AVC video coding standard, high compression rate is obtained by adopting FSBME. Figure.2 illustrates the principle of block matching motion estimation technique. At first, the video frames are segmented into N×N non-overlapping square blocks. Every block within the current frame is matched to the corresponding candidate blocks within a given search range on the previous frame. A matching criterion or distortion function that measures the similarity between the current block and candidate block is calculated. A motion vector for the position of the candidate block, which has the minimum measurement with the current block, is generated to replace the real movement of the objects in a compressed video stream [3]. Thus, the temporal redundancy within a video sequence is reduced.

In this technique, the current block $[C_k]$, where k is the index for the block located at the pixel (x, y), as shown in the Figure. 2, is matched to every candidate block $[R_{k-1}]$ within a (2p+N-1) × (2p+N-1) search window. Here, [-p, p-1] gives the pixel search range. For every candidate block with a displacement (dx, dy), a sum of absolute difference (SAD) is calculated, which is given by the expression (1).

$$SAD = \sum_{m=x}^{x+N-1}\sum_{n=y}^{y+N-1} C_k(m,n) - R_{k-1}(m+dx, n+dy)| \quad (1)$$

C_k(m, n) are the intensity values of the pixels located at position (m, n) in current and previous blocks respectively. Similarly, SAD for the next candidate block is calculated and compared to the existing SAD. The block giving the smaller SAD is kept as the minimum candidate.

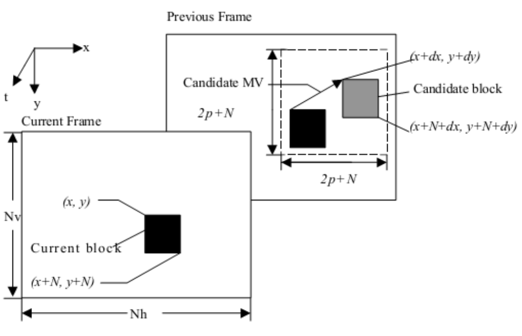


Fig 2. Full Search Block matching motion estimation

This process is repeated until all candidate blocks are matched and a final minimum SAD is obtained. The motion vector is the displacement (dx, dy) of the block, which has minimum SAD within the current block. Thus, full exhaustive search is computationally very intensive and requires the distortion function to eb evaluated many times for each target block to be matched. The total computational complexity (TCC) required for obtaining the best matching block in the reference frame to N X N current block using M search points in a given search window is computed by the expression (2).

$$TCC = [(n1 \times n2).(Sub + Abs + Add)].M \quad (2)$$

In expression (2), n1 and n2 are the rows and columns in the current block. M is the total number of checked points in the search area in the reference frame. These are required for calculating the matching criteria defined by the SAD operations. In general, the computing complexity of the motion vector to each block is $O(n^2)$. Hence, to reduce these computations a method is developed, which drastically reduces the number of computations.

4. PROPOSED TRACE AND OFF-DIAGONAL SUM BASED BLOCK MOTION ESTIMATION

To reduce the computational complexity and to speed up the motion estimation process, we have developed a two-phase motion estimation method that reduces the computations. In linear algebra, the trace (TR) of a square matrix [11] is the sum of all diagonal elements, defined by the expression (3).

$$Trace = \sum_{x=0}^{N-1}\sum_{x=0}^{N-1} C(x,x) --- (3)$$

The trace calculation characterizes the following. It contains, one value from each row dna column of the entire block, and hence no Row or Column is omitted in the block matching. Also the diagonal values considered for trace calculation are position wise linear, and so the motion of pixels are uniform. However, further reduction in number of SAD calculations is done by taking sum of off-diagonal elements so that two pixels in each row and column are compared to improve accuracy in finding nearly matching block. The computing time for trace and off-diagonal is O(2n), and for full search it is $O(n^2)$. Therefore, equation (1) is modified as expression (4) and expression (5), to calculate trace and off-diagonal sum (ODS) elements respectively [20].

$$SAD_{TR(dx,dy)} = \sum_{m=x}^{x+N-1}\sum_{n=y}^{y+N-1} |C_k(m,m) - R_{k-1}(m+dx, m+dy)| \quad (4)$$

And

$$SAD_{ODS(dx,dy)} = \sum_{x+N-1}^{m=x}\sum_{y+N-1}^{n=y} |C_k(m,m) - R_{k-1}(m+dx, m+dy)| \quad (5)$$

The traces (TR) of absolute difference of diagonal pixels of matching blocks of current and ecnerefer frames are calculated by both expressions (5) and (6). The value is used as a deciding factor to eliminate a block or to re-compute the full SAD for the best match in the second phase. Once the matching block for the current frame is found based on the traces match comparison, in the second step, it is verified and confirmed for the best match by calculating the full SAD using expression (1).Thus, the TCC equation 2 is modified as equation (6).

$$TCC = [(2n).(Sub + Abs + Add)].M/4 \quad (6)$$

4.1 Trace Match based Full Search

In the first phase of experimentation, only trace (TR) is used to find the nearly matched block and if the trace values are equal then the SAD value is calculated to find the best match. It is found that the number of SAD calculation is reduced by almost 75%. The process of trace match is depicted in figure 3. Consider a typical macro block of size 4X4 with the gray values of foreman video as shown in Figure 3. The candidate block is selected from the search window size 5X5, with p=2. In the first phase, the trace of current block (TRB) and trace of the candidate block [TRR] are calculated using equation (5). Figure 3 is having pixel values from the search window starting with position P(0,0). The TCB= 802 is compared with TCR of candidate blocks. The TRB and TRR values are equal only at the position p [-1,0] and in all others positions TCR values are different. Hence, only one SAD value is calculated for the position P[-1,0] and for others SAD calculations are skipped. Sometimes TRR and TRB don't match, in such cases, we use off- diagonal sum to find the matching block.

Current Block Trace=802 (a)				TR(candidate blocks)at diff positions of SW (b)			SAD Values (c)		
213	216	216	213	856	851	837			
212	212	212	214	802	872	778	0	7 2	
218	216	214	213	746	738	731			
211	195	178	163	Trace					

Fig 3. (a) Current block , (b) Trace values of candidate block, (c) Typical SAD values

4.2. Trace and Off-Diagonal Match based FS

In order to reduce the number of SAD calculations, an off-diagonal sum (ODS) element is taken for comparison to find nearly matched block. The ODS of the current block (ODSC) and candidate block (ODSR) are calculated. The example values of ODSC=852 and ODSR=852 are shown in the figure 4. The ODSR value of a candidate block at position p(-1,0) is exactly matched and SAD value is calculated for best match. It observed that whenever both ODS and trace are matched, a SAD value of zero predicts the best match. So, only one SAD calculation is carried out, thereby computation saving obtained is around 85%. The Figures 4, depict the complete methodology carried out using both trace and ODS. Only one SAD value is calculated, which gives the best match at position P(-1,0).

Current Block				Candidate Block of SW			
213	216	216	213	216	216	213	211
212	212	212	214	212	212	214	216
218	216	214	213	216	214	213	204
211	195	178	163	195	178	163	151

*TRR=Trace value of candidate blocks

*ODSR=Off-diagonal sum of candidate blocks

TRR Values			ODSR values		
856	851	837	861	855	850
802	792	778	852	834	814
746	738	731	777	752	723

SAD Values		
117	104	69
0	72	176
180	155	186

Fig 4. Matching process of trace and off-diagonal sum

The average numbers of SAD calculations are shown in Table 1 for one Motion Block. Pixel values are taken from standard foreman video sequence and search window size taken is $[2p+1]^2$, and p=7. It observed that the number of SAD computations is reduced significantly because of ODS and trace match. An average of 85% reduction in SAD computations is achieved with combination of ODS and TR.

Table 1, Average number of SAD cal per MB

Current block Size(NXN)	No of SAD values in FS	No SAD cal using TR	No SAD cal using ODS	Using both	% of saving
16X16	207	39	38	24.2	85
8X8	216	46	48	32	83

Average numbers of SAD computations for entire frame with full search with Trace Match(TM) and without trace match are shown in Table 2. From the Tables 1 and 2, we conclude that the proposed method has reduced the number SAD computations drastically. Hence, the computation time to estimate the motion vector is reduced by three to four times and the method is faster than the conventional Full Search.

Table 2. Trace and ODS computations for Full search.

Video	Block size	No. of SAD Computations	
		FS	FS+TM
Caltrain	8X8	403920	82465
	16X16	200980	62187
Foreman	8X8	403920	72867
	16X16	200980	62470
Weather	8X8	403920	86356
	16X16	200980	78920

Algorithm 1: A Two-phase Trace match and SAD Computations

Input: Video file, current frame and reference frame,

Output : Motion Vectors

Descriptions: Frame size-:372X288, Search window size = [2p+1]2, p=7, and block size NXN;

Start,

Step 1: Read the video frames; convert the color frames into gray scale frames.

Step 2: Divide the current frame into macro blocks of certain size (typically 16X16 or 8X8), and select the search window size in the reference frame and set p=7.

Step 3: Calculate the trace and ODS of current block and candidate block.

Step 4: Perform trace match,

Step 4.1 If (trace of the two blocks ≤ specified threshold value) then

go to Step 5,

else

go to step 3.

Step 5: Compute the full SAD

Step 6: Find the minimum SAD.

Step 6.1 If (minimum SAD obtained)

goto step 7

else

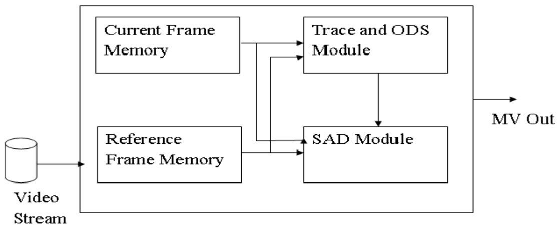
go to step 3 and repeat for the next block.

Step 7: Calculate the Motion Vector.

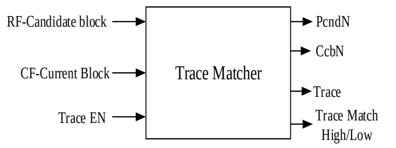
Stop;

5. HARDWARE IMPLEMENTATION OF TRACE AND SAD UNIT

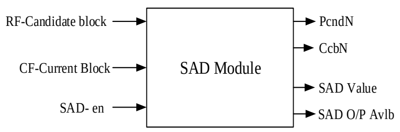
The block diagrams of hardware implementation of motion estimation unit along with trace and off-diagonal sum is as shown in the figure 5



(a)



(b)



(c)

Fig.5 Blocks in hardware implementation a) Motion estimation module. (b) Trace module (c). SAD Module.

Proposed methodology is implemented using Matlab 7.0 and VHDL. The test benches are developed to verify the functionality of each unit. Synthesis is done on latest devices to measure the performances.

The video stream is fed into two separate memories, current frame memory and reference memory. The PcndN and CcbN signals, give the index number of candidate block and current block respectively. The TR and ODS values are calculated using trace module and only on trace match and the SAD module is enabled. In trace module, trace match signal indicates the trace values and the match. The SAD O/P Avlb high indicates that the SAD output is ready and gives the motion vector. It is observed that the SAD module is enabled for only 20% of the calculations, when compared to FS. Hence, the power consumption is reduced.

According to synthesis report, the worst case delay is around 520ns to get the final output and the power consumption is reduced by 15% to 20%. The VHDL module is synthesized by using Spartan-6 and virtex-5 to check the performances. The parameters like area and the maximum frequency of operations are given in Table 3.

Table 3. Device Utilization

straP	seciveD	
	6-natrapS	Virtex-5
fo oN)eludoM ecarT (sBLC	238	184
fo oN)tinU DAS (sBLC	3872	3182
Controller Unit	856	478
Maximum Frequency	354 MHz	420 MHz

6. RESULTS AND DISCUSSION

The experiments are conducted on the proposed method using various standard test video sequences. The number of computations per motion block, average search points, average execution time (CPU speed only), speed up and PSNR are considered as parameters for the comparison against FSBME. The proposed method has given good results compared with other techniques, namely, Three Step Search (3SS), Four Step search (4SS), Full search with Trace (FS+TM) and Diamond search (DS). In Figure.6, analysis of different standard techniques on different standard video sequences with trace match and without trace match is given. It observed that the proposed method is more or less as effective as 3SS, 4SS and DS compared to full search (FS) techniques.

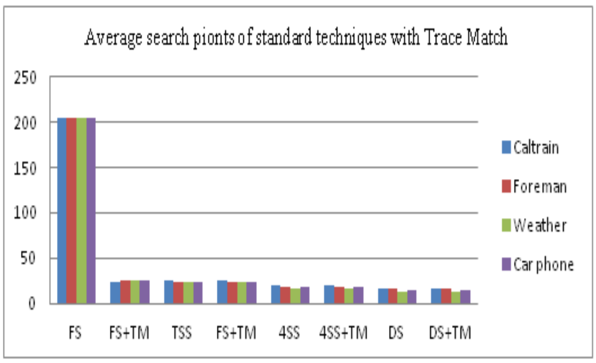


Fig 6. Average of Search Point per MB for Tested ME Techniques with TR.

The analysis is carried out to find the speed up between the standard techniques with TM and without TM and given in

Figure 7,. Speed up is three to four times faster with respect to full search This characteristic is helpful for hardware implementation, because most of the latest standards use FSBME due to its less dependency on previous data and regularity.

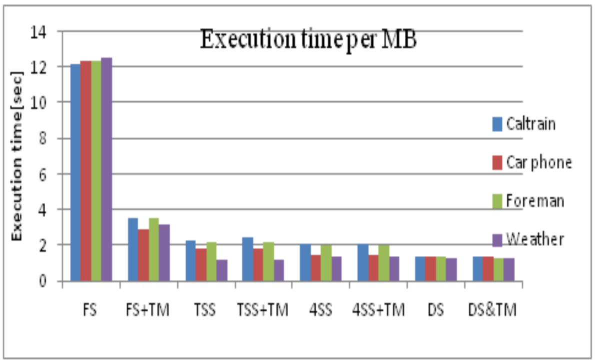


Fig 7. Execution Time (in sec) per Motion Block for tested ME Techniques

6.1. Peak Signal to Noise Ratio

To analyze the video quality obtained from the proposed method, the Peak-Signal-to-Noise-Ratio (PSNR) is considered between pixels of the original frame. Figure 8, gives the PSNR obtained from the proposed method with conventional Full Search(FS), TSS, 4SS and Diamond Search. The PSNR values obtained from the proposed methodology is on par with Full search and fast search technique. Hence, the proposed method assures the better video quality.

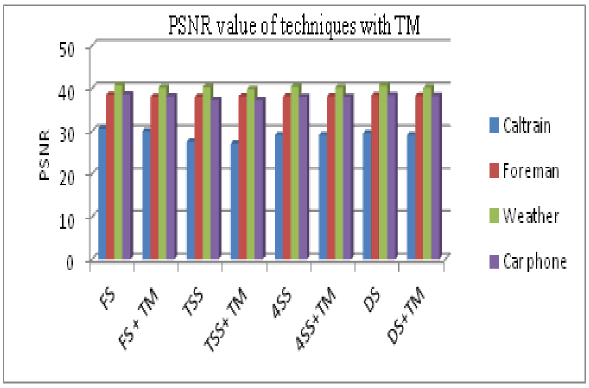


Fig 8. Average values of PSNR obtained on standard videos (dB)

7. CONCLUSION

The combined TR and ODS match based block motion estimation is proposed. Since the Full search (FS) block matching motion search is computationally expensive, in order to minimize the computational overhead, a two-phase methodology is developed. In this method, we have attempted to minimize the number SAD calculations to find the best match. Experimental results show that the proposed methodology is superior in terms of speed-up than the conventional FS block matching. A three to four fold reduction in calculations of SAD, three times speed up is achieved without compromising with PSNR. The FPGA implementation of TR and SAD calculations is carried out. The hardware can execute up to the maximum clock frequency of 420Mhz with gate cost of 5K .Hence, This work is suitable for mobile applications and can be used fast block motion estimation algorithms.

8. REFERENCES

[1] J. Kim, S. Byun, Y. Kim, B. Ahn, “Fast full search motion estimation technique using early detection of impossible candidate vectors”, IEEE Trans. Signal Process. 50 (2002) 2355–2365.

[2] Y. Chan, W. Siu, “An efficient search strategy for block motion estimation using image features”, IEEE Trans. Image Process. 10 (2001) 1223–1238.

[3] Zhu, S. and K. K. Ma (February, 2000). “A New Diamond Search Algorithm for Fast Block Matching Motion Estimation”. IEEE Transactions Image Processing, 9(2), pp. 287–290.

[3] Zhu, C., Lin, X. and Chau, L.P., 2002. “Hexagon-based search pattern for fast block motion estimation”, IEEE Trans. Circuits Syst. Video Technology, May 2002, vol. 12, pp. 349-355.

[4] L.K.Liu and E.EIG,”A block based gradient descent search technique for fast block motion video coding” IEEE Transaction onCSVT,vol.6.419-422,Aug,1996.

[5] H. Loukil, F. Ghozzi, A. Samet, et al. “Hardware implementation of block matching technique with FPGA technology,” 6th International Conference On Microelectronics, Proceedings, pp. 542–546, 2004.

[6] Brunello, D., Calvagno, D., Mian, G.A. and Rina ldo, R., 2003. “Lossless compression of video using temporal information”, IEEE Trans. Image Process., pp. 132–139.

[7] Y. Chan, W. Siu, “An efficient search strategy for block motion estimation using image features”, IEEE Trans. Image Process. 10 (2001) 1223–1238.

[8] S. Wong, S. Vassiliadis, S. Cotofana, “A Sum of Absolute Differences Implementation in FPGA Hardware,” 28th Euromicro Conference (EUROMICRO’02), pp. 183–188, Dortmund, German y, 2002.

[9] Gao, X.Q., Duanmu, C.J. and Zou, C.R., 2000, “A multilevel successive elimination algorithm for block matching motion estimation”, IEEE Trans. Image Processing , March 2000, vol. 9, pp. 501-504.

[10] Puri, A., Hang, H.M. and Schilling, 1997. “An efficient block matching algorithm for motion compensated coding”, in Proc. IEEE Int. Conf.Acoust., Speech, Signal Processing, pp. 1063–1066

[11] Behard Baumgarrtner, “An Inequality for the trace of absolute matrix product values”, arxiv:21106.618, 2011.

[12] Kuo.Liang ,Shyh-Tih Ma,” Predictive Line search: An Efficient Motion Estimation Algorithm for MPEG-4Encoding Systems on Multimedia processors,IEEE Transaction on CSVT,Vol.13.no1,Jan2003.

[13] Kuo-Liang chung and lung Chun Chang”A new Predictive Search Area Approach for Fast Block Mtotion estimation”,IEEE Transaction on Image processing Vol,12. No. 6, june 2003.

[14] Vasily G. Moshnyaga “A New Computationally adaptive Formulation of Block Matching Motion estimation” ,IEEE Transaction on CSVT,Vol.11,No.1,Jan2001.

[15] Michael Bruing and Wolfgang Niehsen “Fast Full Search Block Matching”, IEEE Transaction on CSVT,Vol.11,No.2,Feb 2001.

[16] H.C Huang and Y.P Hung," Adaptive Early Jump-out Technique for Fast Motion Estimation In Video Coding"Graphical Method and Image Processing, Vol 59,No.6.PP 388-394.Nov,1997.

[17] Somanath Senagupta ,V.S.K Reddy," A Fast and Efficient Predictive Block Matching Motion estimation" IJCNSN, Vol 7,No.12,Dec.2007.

[18] S. Sundaravadivelu, Simon Jeyakumar "An Efficient Motion Estimation Algorithm using Trace Match for Fast Video Compression "European Journal of Scientific Research ISSN 1450-216X Vol.53 No.4 (2011), pp.546-554.

[19] Linganagoud Kulkarni, Manu TM, Basavaraj S Anami, "A Two Step Methodology for minimization computational overhead on FSBME", International Journal of u-and e-Service, Science and TechnologyVol.7,No.4(2014),pp.339-348 http://dx.doi.org/10.14257/ijunnesst.2014.7.4.31