ARCHITECTURE DESIGN OF ILLUMINATION CHANGE PROGRESSIVE MOTION ESTIMATION ALGORITHM

Madhuri Bamankar
bamankarmadhuri@gmail.com

P.Muralidhar
pmurali@nitw.ac.in

C.B.Ramarao
cbrr@nitw.ac.in

National Institute of Technology, Warangal-506004, India.

***Abstract*: Some factors of environment and user behavior always dominate the captured video quality (i.e. brightness variation). The full search method cannot give better PSNR for brightness affected video. The brightness correction method improves the PSNR (i.e. quality) of brightness affected video. Architecture design of brightness correction method is proposed in this paper. This hybrid and pipelined architecture is fast and real time implementable.**

***Keywords*: Full search method, motion estimation, brightness correction methods, multi view video coding, H.264/AVC.**

I. INTRODUCTION

Multi view video is captured synchronously by more than one camera in different view angles. Multi view videos are used in 3D and in gaming. While capturing the multi view video, some of frames get affected by brightness changes because of the different view angles of different cameras. In video compression current frame compression depends on previous frame. If previous frame or current frame is affected by brightness then motion estimation method cannot perfect match of current frame microblock into previous frame search range.

There are many brightness correction methods present. General brightness correction is not applied in consumer electronics because it may significantly change the brightness of the original image and can cause annoying artifacts. Kim [2] proposed bi-histogram equalization (BBHE) to meet the requirements of brightness preservation. In Chen [3], the proposed recursive mean-separation histogram equalization (RMSHE) improved the performance of BBHE by using recursive mean-separation. Through the separation of the histogram, the mapping value of the global mean can be modified with these methods to change the curve sharpness of the cumulative density function (CDF).

BBHE and RMSHE can provide an adaptability of global image enhancement to satisfy each different requirement.(MMBEBHE) presents maximum brightness preservation through the estimation of the minimum absolute mean brightness error in [4]. Wang [5] proposed brightness preserving histogram equalization with maximum entropy (BPHEME) to target the specified histogram that can maximize the entropy with some definition of brightness constraint.

The above methods request multiple separation iterations, such that greater hardware complexity is required to implement them.

The illumination change adaptive motion estimation algorithm (ICAME) removes the brightness variation. This algorithm calculates mean of block sum difference of current microblock and reference microblock. This mean gives the average change in pixel values of current microblock compared to neighboring frame microblock. This mean is removed from the SAD of current microblock. This SAD is called as mean removed SAD (MR_SAD).

$$\text{MR\_SAD}=\sum_{x=1}^{M}\sum_{y=1}^{N}|\,\mathrm{C}(x,y)-\mathrm{R}(x-mx,y-my)-\mathrm{DV}| \quad (1)$$

$$\mathrm{DV}=\sum_{i=0}^{N}\sum_{j=0}^{N}\mathrm{c}(i,j)-\sum_{i=0}^{N}\sum_{j=0}^{N}\mathrm{R}(i+x,j+y) \quad (2)$$

DV is difference value. The main problem of ICAME [5] is the complex computation of block mean, which affect the data flow of BMA because the ICAME require mean removable algorithm. The algorithm used in this paper is Illumination change progressive motion estimation.

The rest of the paper is arranged as follows. Section II explains about illumination change progressive motion estimation algorithm. Section III defines proposed architecture design of ICPME algorithm. Simulation results are given in section IV and Conclusion in section VI.

II. ILLUMINATION CHANGE PROGRESSIVE MOTION ESTIMATION

Illumination change progressive motion estimation (ICPME) gives same performance as of Illumination change adaptive motion estimation (ICAME). But ICPME is hardware friendly algorithm. Reference microblock shifts to right or to down by one pixel to give new reference microblock within the search range area. So; actually only one new column or row has to be added and first column or row has to be removed from old microblock. This can make block sum calculation of reference block as progressive block sum. So; DV becomes progressive difference value i.e. PDV.

And SAD becomes the progressive sum of absolute difference (PSAD).

$$PSAD=\sum_{x=1}^{M}\sum_{y=1}^{N} | C(x, y) - R(x - mx, y - my) - PDV| \quad (3)$$

$$PDV(j)= \sum_{n=0}^{j} dsum(n) \;/\; (N \times j) \quad (4)$$

$$dsum(j)= \sum_{i=0}^{N} C(i, j) - \sum_{i=0}^{N} R(i + x, j + y) \quad (5)$$

The PDVIC is obtained by the mean of progressive accumulation for the current *dsum* values. The *dsum* presents the signed difference of current and reference pixel sum with the same line index. The progressive iteration of proposed procedure can achieve a synchronous data flow.

III SIMULATION RESULTS

The ICPME algorithm is simulated in MATLAB. The experiments were conducted on five different video clips. The video clips used are Coastguard (qcif) 176x 144 300 frames, akiyo (qcif), foreman (qcif), mobile (qcif), Highway (cif) 352x288. The quality of recovered video clips was verified by calculating PSNR. Computational complexity (CC) was also calculated for every video.

$$PSNR(I_t,I_{t+1}) = 10 \log(255^2 / MSE) \quad (6)$$

$$MSE((I_t,I_{t+1}))=(1/MN)\sum_{m=1}^{M}\sum_{n=1}^{N} I_t(m, n) - I_{t-1}(m, n) \quad (7)$$

The Illumination change progressive motion estimation algorithm (ICPME) results are compared with Full search block matching without brightness correction algorithm (FSBMA) results. Comparison shows that PSNR of ICPME is much more improved than FSBMA.

Five input video are simulated using MATLAB. Brightness of one of the frame of each video is changed by adding DC value to every pixel of that frame. First is foreman fig(1) video input qcif (172 x 144). Second is coastguard fig(2) video input qcif of size (172 x144) . Third input video is Akiyo cif of size (352x 288). Fourth is mobile qcif of size (172 x 144) and last one is highway cif of size (352 x 288).

Fig(1) and Fig(2) shows 3^{rd} frame of foreman video clip and coastguard 6^{th} frame , affected by brightness change and same reconstructed frame without brightness correction and with brightness correction.

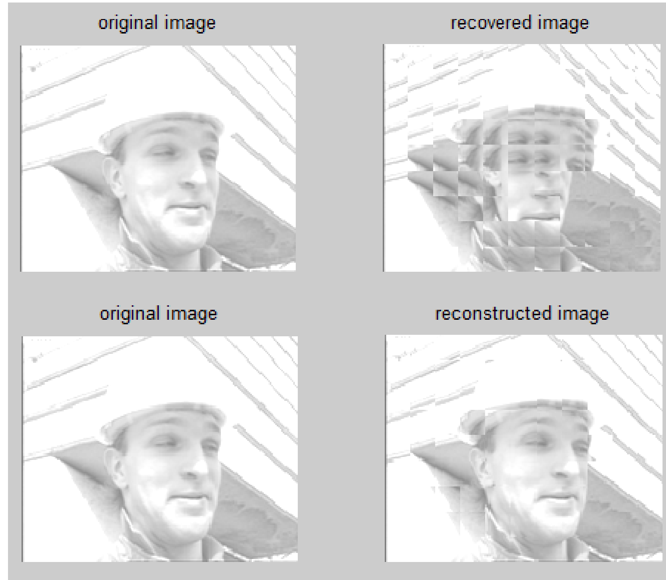

Fig1: Foreman frame without brightness correction and with brightness correction

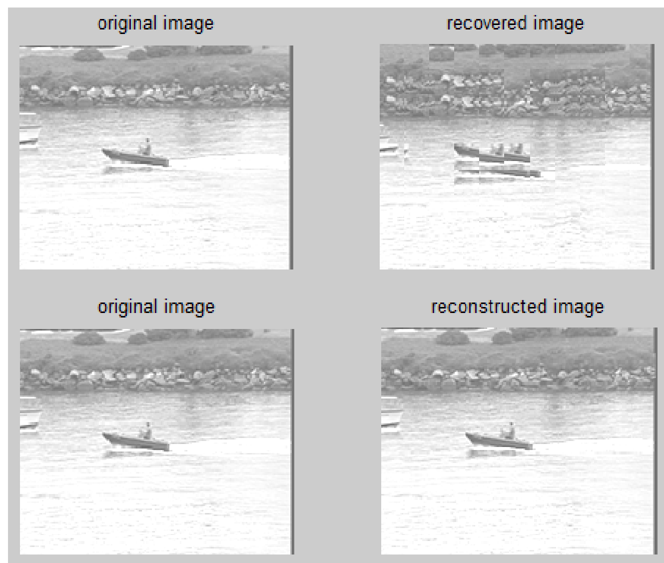

Fig 2: Coastguard frame without brightness correction and with brightness correction

These figures shows that ICPME gives better performance than full search motion estimation (FSBMA). PSNR is also used as one of performance criteria.

Table shown below gives the PSNR comparison of FSBMA and ICPME algorithms.

Method	PSNR				
	Video Input				
	Coastg -uard	Akiyo	Fore-man	Mobile	Highway
FS	8.062	8.130	8.062	7.927	8.123
ICPME	26.16	28.8	15.67	17.97	13.58

Table I: PSNR comparison table of FSBMA an ICPME

IV. ARCHITECTURE

The main blocks of proposed architecture are block sum logic, Processing element array (PEA), Adder tree and motion vector calculation (MV). The block diagram of FFSBMA is given below.

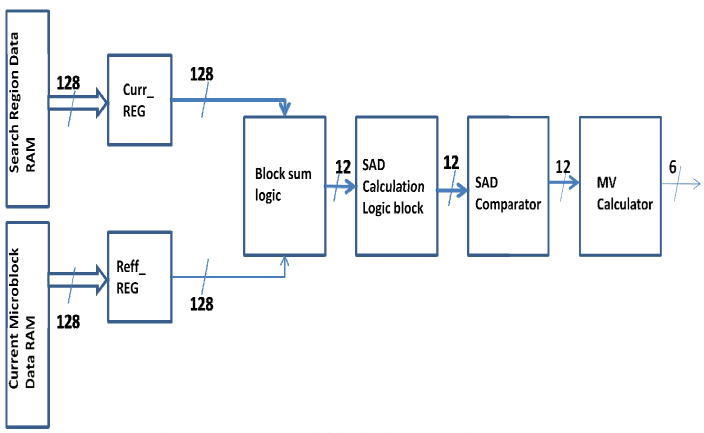

Fig. (3) : Functional block diagram of ICPME

A. Block sum logic

The block sum logic is implemented by using one adder, one subtractor and four 12 bit shift registers. Column adder adds individual column. This column sum is stored into shift register 1 and then shifted to next register. Shift register 4 contains the first column summation and register 1 contains last column summation. Subtrator removes first column summation from block sum. Adder adds new column sum into previous blocksum.

This gives progressive block sum of current and reference microblock. Below figure(4) gives block sum for current microblock. Same logic is applied for reference microblock.

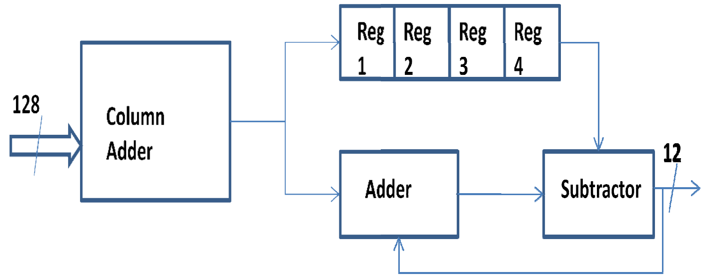

Fig.4: Block sum logic block diagram

B. Processing element array (PEA)

SAD calculation block consists of PEA array.PEA is array of 16 PEs, which are processing parallel. Four PEs PE0, PE4, PE8, PE12 give one 4x4 SAD calculation. All 16 PEs give four 4x4 SADs.

Current RAM gives four pixels at a time. Input current RAM data is constant for four clock cycles. Input Reference data is constant for 3 clock cycles.

Current data flows horizontally through PEA. PE0, PE4, PE8, PE12 forward current data to PE1, PE5, PE9, PE13 and so on. Reference data flows vertically i.e from PE1 to PE4, PE2 to PE5, PE3 to PE6 and so on. PE7, PE11, PE15 require extra 3 reference pixels. It achieves 100% PE utilization by employing a preload register and a search data buffer.

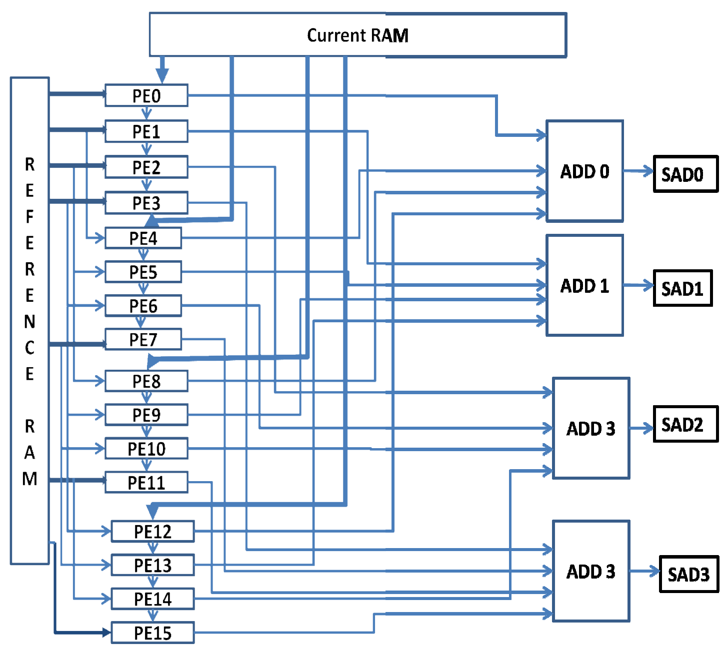

Fig.5 2D systolic array of Processing element array

Such 16 PEAS should be connected in pipelined manner. The vertical data of one PEA comes from horizontal data of above PEA. Top most PEAs get vertical data from reference RAM. 16 PEAs work on 4 microblock SAD calculation in pipelined manner.

C. Processing Element (PE)

PE is the basic block of PEZ array.A PE plays a vital role in processing three different tasks in this design.

- PE computes the absolute difference between the CMD and the SRD.
- Propagates the CMD and SRD values to the next PE (PCMD and PSRD).
- Finally the PE sums up the difference values (DV) of the current and the previous data.

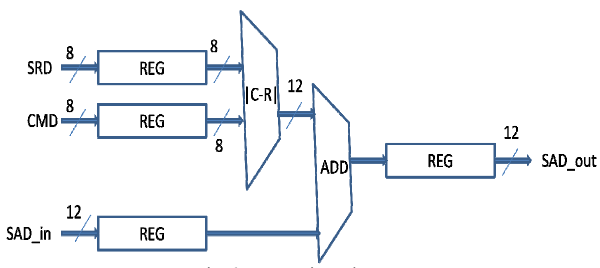

Fig.6 Processing element

V. SYNTHESIS REPORT

This proposed architecture is synthesized on Xilinx ISE 14.2 simulator. Device used for synthesis is (family Vertex 6) xc6vlx75t-3ff484. The source utilization is given below.

Slice Logic Utilization:

Number of Slice Registers:	13265	out of	93120	14%
Number of Slice LUTs:	17354	out of	46560	37%
Number used as Logic:	13130	out of	46560	28%
Number used as Memory:	4224	out of	16720	25%
Number used as SRL:	4224			

Slice Logic Distribution:

Number of LUT Flip Flop pairs used: 23519
Number with an unused Flip Flop: 10254 out of 23519 43%
Number with an unused LUT: 6165 out of 23519 26%
Number of fully used LUT-FF pairs: 7100 out of 23519 30%

Clock period: 7.496ns (frequency: 133.396MHz)

VI. CONCLUSION

Video frames get affected by lighting change during shooting. This degrades the quality of compressed video. IPCME neglects the DC component added into frames while calculating SADs. This improves the quality of compressed video. But this does not removes the DC component from frames.

The proposed architecture for IPCME is Pipelined. Architecture is simple and repetitive. It reuses the data so it supports the progressive calculations.

VI REFERENCES

[1]. Hsueh-Yen Yang, Yu-Ching Lee, Yu-Cheng Fan, Hen-Wai Taso, "A Novel algorithm of local contrast enhancement for medical image," IEEE Nuclear Science Symposium, IEEE NSS, Oct. 26-Nov. 3, 2007.

[2]. Y.-T. Kim, "Contrast enhancement using brightness preserving bihistogram equalization," IEEE Transactions on Consumer Electronics, vol. 43, no. 1, Feb.1997

[3]. Soong-Der Chen, Adb. Rahman Ramli, "Contrast Enhancement using Recursive Mean-Separat Histogram Equalization for Scalable Brightness Preservation," IEEE Trans. on Consumer Electronics, vol. 49, no. 4, Nov 2003

[4]. Yafei Tian, Qingtao Wan, Fengjun We, "Local Histogram Equalization Based on the Minimum Brightness Error," IEEE International Conference on Imaging Graphics, ICIG 2007

[5]. Chao Wang, Zhongfu Ye, "Brightness Preseving Histogram Equalization with Maximum Entropy," IEEE Trans On Consumer Electonics, Vol. 51, No.4, Nov. 2005

[6]. Smolic, K. Muelller, N. Stefanoski, J. Ostermann, A. Gotchev, G.-B. Akar, G. Triantafullidis, A. Koz, "Coding algorithm for 3DTV- a survey," IEEE Trans. On Circuit and Syst. For video technology, vol. 17, no. 11, Nov. 2007.

[7]. U. Fecker, M. Barkowsky, A. Kaup, "Histogram-Based Prefiltering for Luminance and Chrominance Compensation of Multiview Video," IEEE Trans. On Circuits Syst. For Video Technol., Vol. 18, no. 9, Sep. 2008.

[8]. H. K. Cheung, W. C. Sju, D. Feng, T. Cai, "New Block-based Motion Estimation for Sequences with Brightness Variation and Its Application to Static Sprite Generation for Video Compression," IEEE Trans. On Circuits Syst. For Video Technol., Vol. 18, no. 4, Apr. 2008.

[9]. Y. Shen, D. Zhang, C. Huang, J. Li, "Adaptive Weighted Prediction in Video Coding," IEEE Int. Conf. on Multimedia and Expo, ICME, 2004.

[10]. J.-H. Hur, S. Cho, Y.-L. Lee, "Adaptive Local Illumination Change Compensation Method for H.264/AVC Based Multiview Video Coding," IEEE Trans. On Circuits Syst. For Video Technol., vol. 17, no. 11, Nov. 2007.

[11]. A. Vetro, P. Pandit, H. Kimata, A. Smolic,"Joint Mulitiview Video Model (JMVM) 4.0," Joint Video Team of ISO/IEC MPEG and ITU-T VCEG JVT-W207, Apr. 2007. http://www.stereomaker.net/sample/index.html

[12]. K.-M. Yang, M.-T. Sun, L. Wu, "A family of VLSI design for the motion compensation block matching algorithm," IEEE Transactions on Circuits and Systems, vol. 36, no. 10, October 1989.