Hardware-Oriented Adaptive Multi-resolution Motion Estimation Algorithm and Its VLSI Architecture

Guoqing Xiang, Huizhu Jia*, Jie Liu, Yuan Li, Xiaodong Xie
EECS of Peking University
Beijing, 100871
P.R. China
Email: {gqxiang, hzjia, liuzimin, yuanli, xdxie}@jdl.ac.cn

Abstract—In this paper, we propose a hardware architecture of an adaptive multi-resolution motion estimation algorithm (AMMEA) for high definition video encoder to reduce hardware cost. The texture-based search strategies are based on temporal stationarity and spatial homogeneity with Sobel edge operator. The proposed algorithm makes motion estimation more concise. We also propose Sobel edge operator hardware architecture. The four-pixel SAD unit which is the basic processing element (PE) in our proposed architecture is used for SAD calculation and Sobel edge operator computation. The hardware architecture achieves very high data utilization and data throughout. Using our proposed AMMEA with regular data flow, simulation results show that the proposed architecture can significantly reduce the hardware cost with a negligible PSNR loss of 0.03dB compared with the full-search. The design is implemented with SMIC 0.18μm CMOS technology and costs 950K gates count, and it supports the real-time encoding of 1080P@30fps with two reference frames under a clock frequency of 150MHz.

Keywords—Multi-resolution motion estimation; adaptive search strategies; VLSI; data re-use; Sobel

I. INTRODUCTION

Motion estimation (ME) is the most complex part of most popular video compression standards such as MPEG-1/2/4 and H.264/AVC [1]. The goal of integer motion estimation is to reduce temporal redundancies between the current frame and the reference frame. These video coding standards also use new techniques such as variable block size motion estimation (VBSME) and multiple reference frames. Therefore, real-time motion estimation implementation for high definition (HD) video encoder brings great challenges for hardware resources and power consumption.

There are many fast motion estimation algorithms proposed for video coding, such as SEA [2] and DSA [3].Although these software-oriented algorithms achieve time saving, the irregular data flow makes these algorithms unsuitable for hardware implementation. A commonly used hardware-friendly ME algorithm is full-search block matching algorithm (FSBMA) [4], which examines all points in search window. Due to the large search window size requirement for HD video encoder, on-chip memory consumption and computational resource cost are huge. Multi-resolution ME algorithm (MMEA) is a good choice for VLSI implementation to achieve good balance between performance and complexity in HD encoder [5][6], which is developed with a coarse-to-fine search hierarchy. However, as analyzed in [7], traditional MMEA only used fixed search range and the same down-sampling rate for all sequences without discrimination is a big problem. For some sequences with complex texture, wrong motion vector from the coarse level will mislead the search in the fine level which leads to performance degradation. And, for some stationary regions such as background, MMEA also searches in unnecessarily large search window and thus wastes much computational resources.

In this paper, a hardware oriented adaptive multi-resolution motion estimation algorithm (AMMEA) and its VLSI architecture are proposed. AMMEA makes search strategies customized for each block based on stationary and homogeneous features of current macroblock (MB). The four-pixel SAD unit which is the basic PE is applied to search strategy determination and multi-resolution motion estimation. The proposed architecture can make full use of PEs between SAD calculation and Sobel edge operator computation. It will achieve a better balance between the hardware resource and performance.

The remainder of this paper is arranged as follows. Section II gives a brief introduction of hardware oriented AMMEA. The overall VLSI architecture is developed and a search strategy decision block is proposed based on reconfigurable PE array in Section III. The simulation results and the comparisons with other previous works are given in Section IV. Finally, conclusions are drawn in Section V.

II. ADAPTIVE MULTI-RESOLUTION MOTION ESTIMATION ALGORITHM

In this section, we will introduce the adaptive multi-resolution motion estimation algorithm [7]. In traditional three-level MMEA [5][6], it is developed with a coarse-to-fine hierarchical search. In the coarsest level, potential match candidates in the reference are obtained from the largest search window. In the modestly coarse level, several search windows are centered at the candidates provided from the immediate upper coarse level. In the finest level, the search range (SR) will be set to be very small for calculation reduction and is

*The corresponding author， Huizhu Jia is with Peking University, also with Cooperative Medianet Innovation Center and Beida (Binhai) Information Research.

centered at a single candidate selected from modestly coarse level. However, the proposed AMMEA doesn't use a fixed search range and the same down-sampling. For different sequences, adaptive search strategies are adopted to have a better balance between computational expense and performance. The spatial homogeneity and temporal stationarity characteristics of video sequences are detected to adaptively determine search strategies.

The details of the procedure to determine search strategies are described as follows:

1) Stationary Region Determination. Stationary regions are considered as static regions in the temporal dimension. Stationary regions are more likely located in a very small region. Therefore, it is reasonable to search around the origin (0, 0) with a small range. Stationary regions are detected by temporal information. The difference between current and reference MB can be computed by (1). Where C[i, j] and P[i, j] represent the luminance values in the current and reference MB located at origin (0, 0), respectively. If the Diff is smaller than T, the current MB is considered as stationary MB.

$$\mathrm{Diff} = \sum_{i=1,j=1}^{16,16} abs(C[i,j] - P[i,j]) \tag{1}$$

2) Homogeneous Region Determination. If textures of a region have similar spatial property, the region is defined as homogeneous region. Texture complexity can be represented by edge information. Sobel edge operators will be adopted to detect homogeneous region. For a pixel in a luma picture, we define the corresponding edge vector, $\vec{D}_{i,j} = \{dx_{i,j}, dy_{i,j}\}$ as

$$\begin{aligned} dx_{i,j} &= p_{i-1,j+1} + 2\times p_{i,j+1} + p_{i+1,j+1} - p_{i-1,j-1} - 2\times p_{i,j-1} - p_{i+1,j-1} \\ dy_{i,j} &= p_{i+1,j-1} + 2\times p_{i+1,j} + p_{i+1,j+1} - p_{i-1,j-1} - 2\times p_{i-1,j} - p_{i-1,j+1} \end{aligned} \tag{2}$$

where *dx* and *dy* represent the degree of difference in vertical and horizontal directions. Therefore, the amplitude of the edge vector can be computed by

$$\mathrm{Amp}(\vec{D}_{i,j}) = |dx_{i,j}| + |dy_{i,j}| \tag{3}$$

The homogeneity size is the same as the current block size (16x16). The r and c indicate the row and column of the block, respectively. The sum amplitude of one MB is represented as

$$\mathrm{H}(r,c) = \sum_{i,j \in N\times N} \mathrm{Amp}(\vec{D}_{i,j}) \tag{4}$$

3) Search Strategies Determination. Fig.1 shows the AMMEA flow chart. First, we conduct Stationary Region Detection with the method discussed in 1). If a current MB is detected as stationary MB, the proposed algorithm adopts the search strategy I. Then, if current MB is not a stationary MB, Homogeneous Region Determination will be performed. The sum of the amplitude of the edge vectors in the block can be obtained by (4), and it is divided into three categories by two thresholds, Thd1 and Thd2. Three categories correspond to different search strategies. We different sequences under different resolutions to determine the two threshold. According to exhaustive experiments on various video sequences, Thd1=20000 and Thd2=16000 will achieve relatively good performance. As analyzed above, the details are shown in Table I.

III. PROPOSED VLSI ARCHITECTURE

Since large search window is absolutely necessary for HD video encoder, high processing throughput is the largest challenge in IME hardware implementation. In our architecture, the search window of [-128, 128)×[-96, 96) is adopted. It presents a significant challenge in external memory bandwidth, data latency and computation resources. A hardware-efficient VLSI architecture for AMMEA based on PE array reuse for SAD and search strategy decision calculation can conquer these problems.

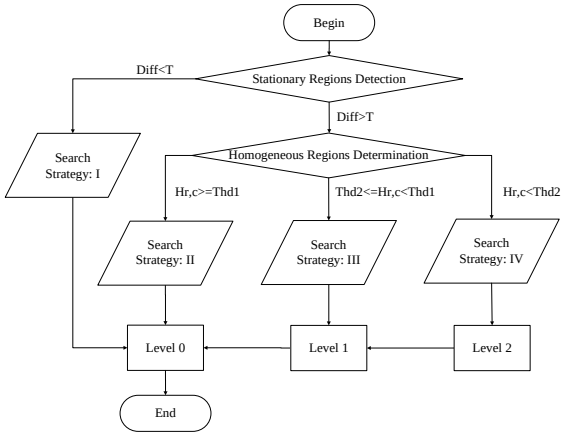

Fig. 1. AMMEA flow chart

TABLE I DIFFERENT SEARCH STRATEGY

Search Strategy / Level		IV	III	II	I
Level2	SR	$SRx = 128$ $SRy = 96$	OFF	OFF	OFF
	Center	(0,0)			
	Sampling	16:1			
Level1	SR	$SRx^{IV}_{L1} = 8$ $SRy^{IV}_{L1} = 8$	$SRx^{III}_{L1} = 16$ $SRy^{III}_{L1} = 16$	OFF	OFF
	Center	3 bestMVfrom lev2 & PMV	PMV		
	Sampling	4:1	4:1		
Level0	SR	$SRx^{IV}_{L0} = 4$ $SRy^{IV}_{L0} = 4$	$SRx^{III}_{L0} = 8$ $SRy^{III}_{L0} = 8$	$SRx^{II}_{L0} = 12$ $SRy^{II}_{L0} = 12$	$SRx^{I}_{L0} = 12$ $SRy^{I}_{L0} = 12$
	Center	1 best MV from lev1	1 best MV from level1	PMV	(0,0)
	Sampling	NO	NO	NO	NO

A. Overall Architecture

The proposed architecture is block-level pipelined and level C+ data reuse scheme [8] which requires zigzag coding pattern is adopted in our design to reduce external memory bandwidth. As analyzed in Section II, we proposed the VLSI architecture based on AMMEA. The top-level block diagram of one search engine of the proposed architecture is shown in Fig. 2. It comprises three types of main function modules: control, computation and storage modules. Control modules mainly perform search strategy decision, stationary decision, homogeneous decision and three levels FSM, etc. The Sobel operator computation and SAD reuse PE array and the details will be further discussed in Section III-B. According to three-level AMMEA, the largest search window is divided into 16 sub-windows at level 2. In [6], 16 sub-windows are mapped into 16 memory units. Each unit consists of ram_even and ram_odd. Fig. 3. shows the four-pixel SAD unit as the basic PE in our proposed architecture. All computation of SADs at each

level and search strategy determination can be calculated by the four-pixel PE architecture. A data multiplexer (MUX) based on this PE architecture is used for the calculation of the amplitude of edge vector and SAD calculation. Considering two reference frames used in our encoder, there are totally 64×2 parallel four-pixel PEs in our proposed architecture.

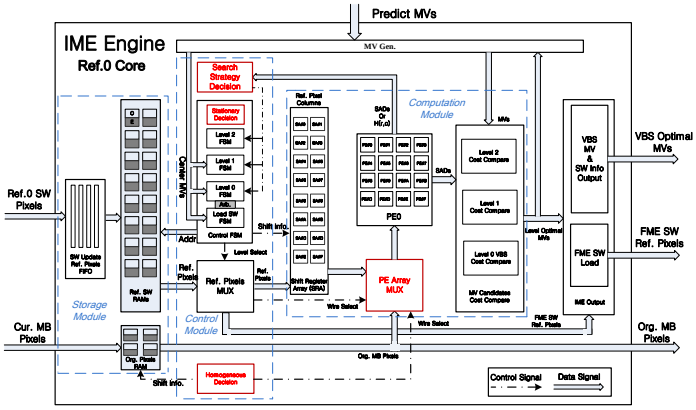

Fig. 2. One search block diagram of the proposed architecture

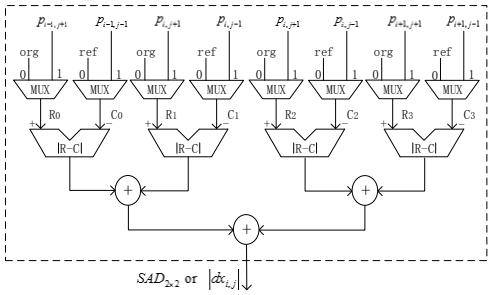

Fig. 3. Four-pixel PE architecture

B. *Search Strategy Decision*

In Search Strategy Decision module, PE arrays calculate a current MB Diff by (1) and H(r, c) by (4) first, then, it will determine which search strategies can be used, as illustrated in Fig. 1. Configuration parameters which are shown in Table I will be transferred to each level's FSM. Finally, motion estimation module will work with the configuration parameters. The detailed search strategy decision procedure will be analyzed below:

Step 1: Stationary MB Decision. As discussed in Section II, the method for detecting stationary region is to calculate the difference between current and reference MB by (1). When computing the difference, the control signal of MUXs will be set to "0" to calculate SAD. The calculation result will be directed to search strategy decision module and compared with threshold T to determine the adopted search strategy.

Step 2: Homogeneous MB Decision. If the difference between current MB and reference MB is greater than threshold T, homogeneous decision module will be in work. Since four-pixel PE architecture can calculate four-pixel SAD, the equation (2) should be changed as

$$dx_{i,j} = (p_{i-1,j+1} - p_{i-1,j-1}) + (p_{i,j+1} - p_{i,j-1}) + (p_{i,j+1} - p_{i,j-1}) + (p_{i+1,j+1} - p_{i+1,j-1}) \quad (5)$$

As shown in Fig. 3, when computing Sobel edge operator by (5), the control signal of MUXs will be set to "1" and is calculated by the four-pixel PE. Every two four-pixel PEs are mapped to support calculating $dx_{i,j}$ and $dy_{i,j}$, respectively. Since there are totally 64 parallel four-pixel PEs in one search path, as illustrate in Fig. 4, all PE arrays can be connected to the 18×4 systolic array for calculating 32 amplitudes of the edge vector defined by (3). Thus, just 8 cycles are needed to calculate all the sum edge amplitude of one MB.

Fig. 4. shows that one original MB pixels are buffered into the 16×16 systolic arrays as the blue circles. Since the proposed architecture of AMMEA is block-level pipelined and the scan mode is zigzag coding pattern, it is difficult to fetch adjacent block pixels. So, in order to calculate the Sobel edge operator of the edge pixels in the MB, it is necessary to pad one pixel in the upper and lower boundary as the yellow circle shown. Meanwhile, the padding pixels of the left and right boundaries are fed back to the edge of original MB pixels as the white circle shown. Therefore, 32 amplitudes of the edge vector can be calculated per cycle and 8 times of upward move are needed to calculate the sum edge amplitude of one MB. The calculation result will be directed to search strategy decision module for search strategy decision.

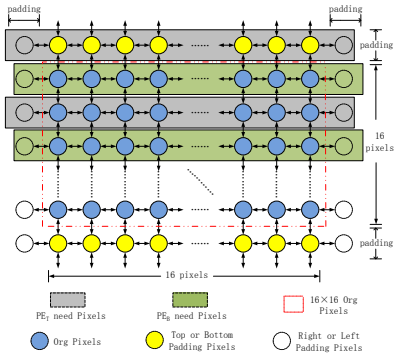

Fig. 4. 16×18 systolic array of current MB

Step 3: Search Strategy Decision. The calculation results from step1 and step2 will be sent into the threshold compare module at first. The best search strategy will be selected by the criterion as discussed in Section I. Configuration parameters will be transferred to each level's FSM and set the control signal of MUXs to "0".

IV. Implementation Results

A. *Performance of the Proposed AMMEA*

The hardware-oriented motion estimation algorithms such as full search block matching algorithm (FSBMA) [2] and MMEA [5][6] are chosen for performance comparison. The proposed algorithm was implemented on AVS Jizhun profile. The test conditions are as follows: 1) Search window of [-128, 128)×[-96, 96), 2) SAD is used as the matching distortion criterion, 3) Reference frame number equals to 2, 4) VLC is enabled, 5) MV resolution is 1/4 pel, 6) GOP structure is IBBPB, 7) Inter block mode from 16x16 to 8x8. Table II shows that the PSNR degradation of proposed algorithm compared with FSBMA. The metrics is BD-PSNR using four QPs (24, 28, 32, 36). The average quality loss of the proposed algorithm

compared with FSBMA is about 0.03dB. The last column of Table II shows where AMMEA achieves better performance than [5][6] for all kinds of test sequences. Especially for the feature-complex sequences such as "Mobile" and "Fireworks", we have higher PSNR gain because search strategy at this region is always searching without down sampling.

TABLE Ⅱ THE PSNR PERFORMANCE COMPARISON

Resolution	Sequence	Proposed (dB)	[5][6] (dB)	Diff (dB)
1080P	Fireworks	-0.10	-0.17	0.07
	BasketballDrive	-0.06	-0.07	0.01
	Tractor	-0.06	-0.07	0.01
	Cactus	-0.01	-0.02	0.01
	MobcalYer	-0.00	-0.01	0.01
	Crowdrun	-0.00	-0.04	0.04
720P	Spincalendar	-0.07	-0.13	0.06
	Sheriff	-0.00	-0.01	0.01
	City	-0.03	-0.07	0.04
	Optis	-0.00	-0.01	0.01
D1	Mobilecalendar	-0.05	-0.14	0.09
	Flowergarden	-0.00	-0.05	0.04
CIF	Mobile	-0.00	-0.18	0.18
	Foreman	-0.04	-0.11	0.07
	Kiel	-0.05	-0.21	0.16
	Crew	-0.02	-0.07	0.05

B. Hardware Implementation Results

The proposed hardware architecture was designed with Verilog-HDL description language and synthesized by Synopsys Design Compiler with SMIC 0.18μm CMOS standard cell library. The design contains about 950K gates and the total size of SRAM is 192KB. The circuit can work at maximum operation frequency of 210 MHz It aims at main profile H.264 and Jizhun profile AVS for high definition video encoder. Under a clock frequency of 150MHz, the proposed architecture supports the real-time encoding of 1080P@30fps with two reference frame simultaneous searches with search window size of 256×192.

Table III shows the detailed hardware cost of the proposed algorithm and the comparison with previous works. Since the calculation of search strategy decision also reuses the four-pixel PE arrays, this module incurs a modest hardware resource increase with just 30K. In Table III, three architectures, AMMEA, [5]/[6] and [10], support the real-time encoding of 1080P@30fps. And, AMMEA is the fastest among them. Compared with [5]/[6], the gate count is close to the proposed architecture, but the required operating frequency for the real-time application for 1080P images is 220 MHz. However, the proposed architecture can reach the throughput of 600 cycles per MB, and 150 MHz clock system frequency is enough for the real-time encoding of 1080P@30fps.

V. CONCLUSION

In this paper, we proposed a hardware-oriented adaptive multi-resolution motion estimation algorithm for hardware HD video encoder implementation. The proposed architecture exhibits its advantages by providing not only better PSNR performance than [5][6] , but also a low latency and high data reuse that is appropriate for VLSI implementation. We also proposed the search strategy decision architecture with four-pixel PE array re-use scheme. The design is implemented with SMIC 0.18μm CMOS technology and costs 950K gate count, and it supports the real-time encoding of 1080P@30fps with two reference frames under a clock frequency of 150MHz.

TABLE Ⅲ HARDWARE COST COMPARISON

Designs	Proposed	MMEA [5][6]	Huang [9]	Liu [10]	Chen [11]	Deng [12]
Video Spec.	1080P@ 30fps	1080P@ 30fps	720P@ 30fps	1080P@ 30fps	720P@ 30fps	SD@ 30fps
Ref.Number	2	2	4	1	1	1
Search Range	256×192	256×192	128×64	196×128	128×64	65×65
Number of PE	512	512	N/A	2048	128×8	16×16
Throughput (Cycles/MB)	600/536	848	N/A	960	1536	5216
On chip Memory(KB)	96×2	96×2	34.72	40 (dual port)	13.71 (dual port)	62
Technology (μm)	0.18	0.18	0.13	0.18	0.18	0.18
Gate Count(K)	475×2	460×2	992.8	486	305	210
Working Frequency(MHz)	150	220	108	200	108	260

ACKNOWLEDGMENTS. This work is partially supported by grants from National High Technology Research and Development Program of China (863 Program) under contract No.2015AA015903, the National Science Foundation of China under contract No.61502013 and No.61520106004

REFERENCES

[1] Draft ITU-T Rec. Final Draft Int. Standard of Joint Video Specification, ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, May 2003.

[2] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," IEEE Trans. Image Process., vol. 9, no. 3, pp. 501 –504, Mar. 2000.

[3] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," IEEE Trans. Image Process., vol. 9, no.2, pp. 287–290, Feb. 2000.

[4] T.C. Chen et al, "Analysis and Architecture Design of an HD720p 30 Frames/s H.264/AVC Encoder," IEEE Trans. Cir. Syst. Video Tech., vol. 16, no. 6, pp. 673-688, June 2006.

[5] X. H. Ji, C. Zhu, H. Z. Jia, X. D. Xie, H. B. Yin, "A Hardware-Efficient Architecture for Multi-Resolution Motion Estimation Using Fully Reconfigurable Processing Element Array,"in Proc. ICME, Jul. 2011, pp. 1–6.

[6] H. B. Yin, H. Z. Jia, H. G. Qi, X. H. Ji, et al, "A Hardware-Efficient Multi-Resolution Block Matching Algorithm and Its VLSI Architecture for High Definition MPEG-Like Video Encoders," IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 9, pp. 1242‑1254, Sept. 2010.

[7] Jie Liu, Xianghu Ji, Chuang Zhu, Huizhu Jia, Xiaodong Xie, WenGao, "Adaptive multiresolution motion estimation using texture-based search strategies," IEEE International Conference of Consumer Electronics., pp. 363-366, Jan. 2014.

[8] C. Y. Chen, C. T. Huang, Y. H. Chen, L. G. Chen, "Level C+ Data Reuse Scheme for Motion Estimation With Corresponding Coding Orders," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 4, pp. 553–558, Apr. 2006.

[9] Y.-W. Huang, T.-C. Chen, et al, "A 1.3 TOPS H.264/AVC single-chip encoder for HDTV applications," in IEEE ISSCC Dig.Tech. Papers, pp128-129, Feb.2005.

[10] Z.Y. Liu, Y. Song, M. Shao, et al, "HDTV 1080P H.264/AVC encoder chip design and performance analysis," IEEE J. Solid-State Circuits, vol. 44, no. 2, 816 pp. 594–608, Feb. 2009.

[11] T.-C. Chen, S.-Y. Chien; Y.-W. Huang, et al, "Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 6, pp. 673–688, Jun. 2006.

[12] L. Deng, W. Gao, M. Z. Hu, Z. Z. Ji, "An efficient hardware implementation for motion estimation of AVC standard," IEEE Trans. Consumer Electron., vol. 51, no. 4, pp. 1360-1366, Nov. 2005