A 995Mpixels/s 0.2nJ/pixel fractional motion estimation architecture in HEVC for Ultra-HD

Gang He, Dajiang Zhou, Zhixiang Chen, Tianruo Zhang and Satoshi Goto
Graduate School of Information Production and Systems, Waseda University, Kitakyushu, Japan

***Abstract*—This paper presents a fractional motion estimation (FME) design in high efficiency video coding (HEVC) for ultra-high definition video (Ultra-HD). To reduce complexity and achieve high throughput, the design is co-optimized in algorithm and hardware architecture. Bilinear quarter pixel approximation, together with a 5T12S search pattern is proposed to reduce the complexity of the interpolation and search process. Furthermore, we introduce an exhaustive size-hadamard transform (ES-HAD), to improve coding quality, and determine the best transform size rather than using complex transform coding. Besides, a data reuse method of ES-HAD is applied to reduce the hardware overhead. This design is implemented in 65nm CMOS chip and verified by FPGA based evaluation system. It achieves 995Mpixels/s for 7680x4320 30fps encoding, at least 4.7 times faster than previous designs. Its power dissipation is 198.6mW at 188MHz, with 0.2nJ/pixel power efficiency. Despite high complexity in HEVC, the chip achieves 56% improvement on power efficiency than previous works in H.264.**

Keywords—FME; HEVC; Ultra-HD; hardware architecture;

I. INTRODUCTION

Recently, the video compression standard called high efficiency video coding (HEVC/H.265), has been approved by Joint Collaborative Team on Video Coding (JCT-VC) [1]. It doubles the data compression ratio compared to the precursor of AVC/H.264. Besides, high resolutions such as ultra-high definition (Ultra-HD 7680x4320, 3840x2160) are supported for next-generation applications.

As an important component of video encoding, fractional motion estimation (FME) provides the sub-pixel refinement. It improves the rate-distortion performance significantly about 3-6dB, but results in high computation complexity (40% encoding time) due to complex interpolation and fractional search process. Previous works [2]-[5] have proposed efficient designs in H.264. Due to the new features in HEVC, most of them cannot be applied directly. Quad-tree structure enables the motion estimation in coding blocks from 64x64 to 8x8. Moreover, the interpolation is processed with complex 7/8-tap filters. In addition, the transform coding has been added as residual quad-tree (RQT).

In this work, an efficient FME architecture in HEVC is proposed. The design targets Ultra-HD 7680x4320 30fps real-time encoding. The co-optimization of algorithm and hardware are characterized as follows. (1) By using bilinear quarter pixel approximation, we reduce 76% interpolation complexity and save transform operation for quarter candidates. (2) A 5T12S search pattern is proposed to achieve a tradeoff between hardware cost and coding quality. 48% hardware cost is reduced with negligible quality loss, compared with conventional 9T25S. (3) Exhaustive size-hadamard transform (ES-HAD) is adopted in FME. It avoids unifying all blocks into small transform ones. Furthermore, it determines the best transform size, rather than using the complex RQT. Besides, data reusing in ES-HAD is exploited and 58% hardware cost is reduced, compared with the straightforward implementation.

The rest of paper is organized as follows. Section II reviews FME algorithm in HEVC referenced software. In section III, we present our proposed algorithm. Hardware architecture is described in Section IV. Implementation results are shown in Section V. Section VI concludes the paper.

II. FME ALGORITHM IN REFERENCED SOFTWARE

In HEVC, an input picture is divided into coding tree blocks (CTBs, typically 64x64) and each CTB can be further divided into coding blocks (CBs, limited to 8x8) through a recursive quad-tree structure. The three symmetric and four asymmetric prediction modes are supported for each CB. For each prediction mode, the procedure of motion estimation is classified into three steps. Firstly, integer motion estimation (IME) performs motion search to find integer motion vectors (IMVs). Next, FME provides the sub-pixel refinements around IMVs. At last, RQT processes the transform coding. The transform sizes are allowed in the range of 4x4 to 32x32 and the best one is decided with rate distortion optimization (RDO). The detailed FME are carried out as follows: the fractional pixels (pels) with quarter accuracy are first interpolated with 7/8-tap filter. Then, two refinements from half pixel to quarter pixel are performed sequentially. In each refinement, nine neighboring points around the former best result are searched. The cost function adopted in FME is the hadamard transform absolute difference (HAD). It includes the operations of difference generation (DG), 4x4/8x8 hadamard transform (HT), absolute value and cost accumulation.

III. PROPOSED FME ALGORITHM

A. Bilinear Quarter pixel Approximation

In HEVC, fractional pixels are classified into a, b and c types and performed with corresponding independent 7/8/7-tap filters. This is defined for motion compensation (MC). If we apply this into motion estimation, it introduces a high complexity interpolation. Moreover, relative to H.264, the operation linearity cannot be utilized and hardware cost increases largely. In H.264, the interpolation adopts a half-then-

978-1-4799-0280-4/13/$31.00

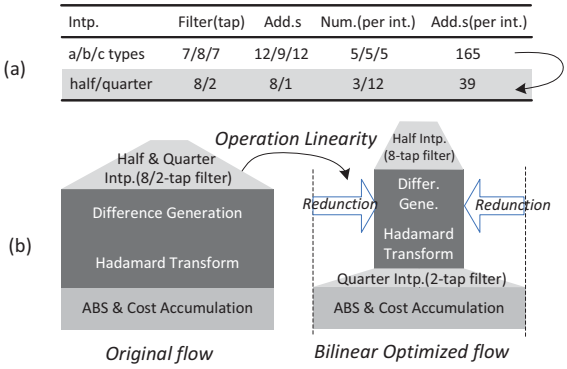


Fig. 1. (a) Additions reduction (b) Saving DG and HT for quarter pixels.

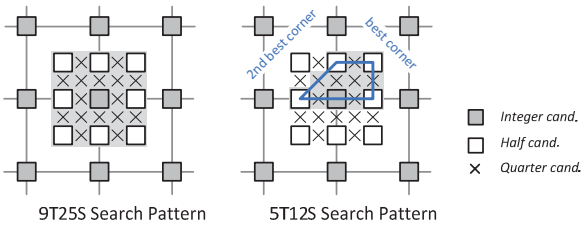


Fig. 2. (a) conventional 9T25S (b) proposed 5T12S search pattern.

quarter model. It first generates half pixels with 6-tap filter, and then uses half pixels to generate quarter ones with a 2-tap bilinear filter. Due to the linearity of DG and HT, the operation results of quarter pixels can be directly calculated from the results of half pixels, with the bilinear filter.

In this design, we propose a bilinear quarter pixel approximation (BQA) scheme. It employs the half-then-quarter model and uses bilinear filter for quarter pixels. The 8-tap filter is still adopted for half pixels to maintain the coding quality. As shown in Fig. 1 (a), the interpolation complexity is reduced about 76%, from 165 to 39 additions. The PSNR is degraded only 0.02dB. Moreover, FME process flow is optimized with the operation linearity, as shown in Fig. 1 (b). We calculate the coefficients of quarter pixels with bilinear 2-tap filter, and the DG and HT operations are saved. It is noted MC is not included here and needs to be handled separately.

B. 5T12S Search Pattern

The search pattern in referenced software takes two iterations, and introduces long latency. One iteration search is suitable and widely used for high throughput hardware design. With one iteration, most existing works focus on the reduction of search candidates (SCs), since they generally determine the hardware cost. However, with proposed bilinear optimized flow shown in Fig. 1 (b), hardware cost is largely dependent upon the number of transformed candidates (TCs), since the transform takes most of the complexity.

The conventional search area of central 5x5 provides high coding quality [5], shown in Fig. 2 (a). Using the bilinear optimized flow, nine candidates are processed with transform among the total 25 ones. It is denoted as 9T25S. To further

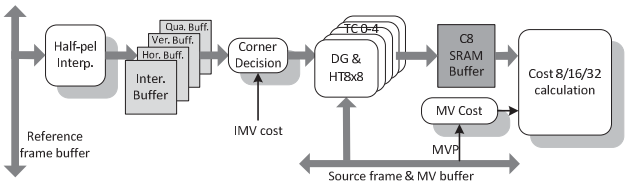


Fig. 3. Block diagram of the FME design

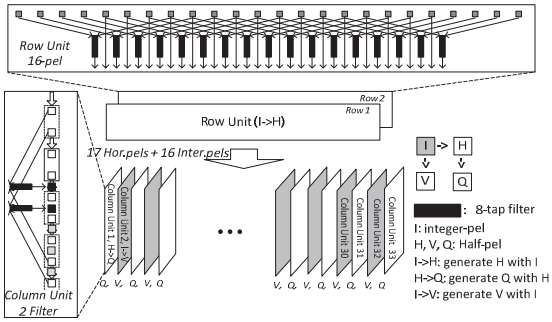


Fig. 4. Block diagram of half-pel interpolation module.

reduce the hardware cost, we propose a 5T12S search pattern. Integer evaluation results are utilized to predict the high possibility area (best corners) for FME. Saving fewer transform candidates and maintaining the coding quality, the search pattern is decided as shown in Fig. 2 (b). Five TCs are processed to generate the results of 12 SCs, denoted as 5T12S. Relative to the 9T25S, it reduces 48% hardware cost with only 0.02dB PSNR degradation.

C. Exhaustive Size-HAD

In HEVC, the residual quadtree (RQT) allows transform sizes in the range of 4x4 to 32x32. The approach enables the adaptation of transform to the varying space-frequency characteristics of the residual. However, the adopted HAD as rate distortion estimation in FME, is limited to 4x4/8x8 transform sizes.

In this design, we apply an exhaustive size HAD (ES-HAD). The residuals are processed by HT8x8, HT16x16, and HT32x32, and then compared recursively to get rate distortion cost. The technique avoids unifying variable blocks into small size blocks and improves the encoding performance for FME, up to 0.1dB video quality. Moreover, it determines the best transform size, with much lower complexity, relative to the complex RDO adopted in RQT. The average PSNR drop is about 0.05dB. In addition, we exploit data reusable features in ES-HAD and reduce hardware cost largely, which is explained in Section 4.2.

IV. Hardware Architecture

Fig. 3 shows the block diagram of proposed FME design. With referenced integer pixels, half fractional pixels are generated by half-pel interpolation module, and then stored in the buffers. After the corner decision, five TCs are processed parallel by five DG&HT8x8 units. The coefficients of HT8x8 (C8) are stored in a SRAM buffer, and reordered to be accessed

$$H_{2x}=\begin{pmatrix} H_x & H_x \\ H_x & -H_x \end{pmatrix} \qquad A=\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

$$\exists\ T_{x-ij}=H_x \bullet A_{ij} \bullet H_x^{\,T} \qquad T_{2x}=H_{2x} \bullet A \bullet H_{2x}^{\,T}:$$

$$T_{2x}=\begin{pmatrix} T_{x-11}+T_{x-12}+T_{x-21}+T_{x-22} & T_{x-11}-T_{x-12}+T_{x-21}-T_{x-22} \\ T_{x-11}+T_{x-12}-T_{x-21}-T_{x-22} & T_{x-11}-T_{x-12}-T_{x-21}+T_{x-22} \end{pmatrix} \quad (1)$$

TABLE I. COMPARISON FOR INDIVIDUAL IMPLEMENTATION AND RESUE IMPLEMENTATION

Addition Layer	HT8x8	HT16x16	HT32x32
Individual	6	8	10
Date Reuse	6	2	2

for the reuse operations of HT16x16 and 32x32. At last, all 12 SCs with ES-HAD costs are generated and compared in cost calculation module.

A. Parallelism

To design the FME architecture with high throughput and minimal hardware cost, the degree of parallelism has to be carefully considered. For the interpolation, an efficient way is to design the processing unit based on the smallest block. Others are decomposed to re-utilize the hardware. Based on the 16x8 block in this design, we use the horizontal unit with 16-pixel parallelism. It generates 17 half horizontal-pels with 8-tap filter and directly outputs 16 inter-pel. The corresponding 33 vertical units are needed to generate half vertical-pels and quarter-pels. However, it has to work at a high frequency 380MHz to achieve targeted throughput. Thus, we double the parallelism by adopting two horizontal units and embedding two 8-tap filters into a vertical unit, as shown in Fig. 4. By doing so, it takes 8 cycles to processes a 16x8 block. Other processing modules are designed based on 5T12S search pattern. The DG&HT8x8 units are duplicated five times and the cost calculation module contains 12 processing units. Being fully pipelined, all these units adopt 16-pixel parallelism, in accordance with the processing speed of interpolation (16x8 block/8cycles). It should be noted, in order to achieve high throughput, we support three selected modes from CB 64x64 to 16x16 by using IME evaluation results, with 0.07dB quality loss. A CTB (64x64) can be decomposed into 32 16x8 blocks. Thus we calculate the total required cycle count per CTB as 3 x 32 x 8 = 768 cycles.

B. Data Reuse in ES-HAD

In this design, ES-HAD is applied, where the residuals are processed with HT 8x8, 16x16 and 32x32. As shown in Table I, with individual implementation, six, eight, ten addition (ADD) layers are needed for pipeline design, respectively. Due to the recursive feature of HT, we exploit the reusable relationship between different sizes. As shown in the derivation (1), the transform calculation for T2x can reuse the transformed result Tx. Only two bufferfly ADD layers are needed. Based on that, we only implement the architecture of HT8x8. Two and two

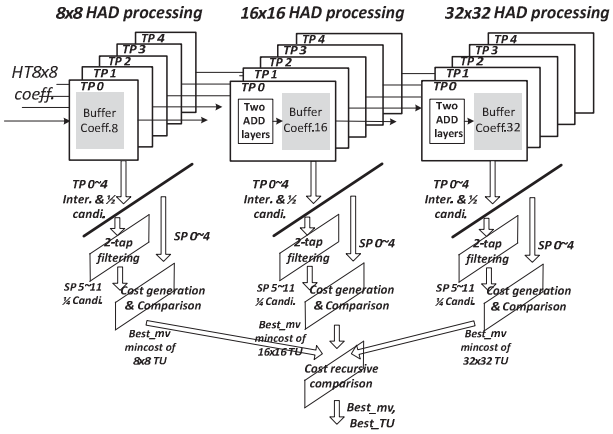


Fig. 5. Block diagram of the cost calculation module.

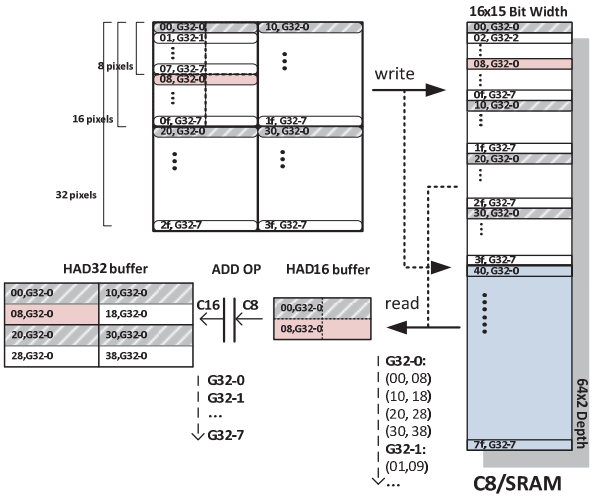
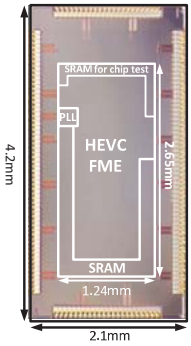


Fig. 6. Organization and access pattern of the on-chip SRAM.

ADD layers are used to calculate HT16x16 and 32x32. Comparing with individual implementation, 58% hardware cost is reduced. Fig. 5 shows the block diagram of proposed cost calculation unit. It contains the processing for 8x8, 16x16, and 32x2 HAD. The coefficients are first used to calculate the RD cost, and also utilized to generate the coefficients of larger transform.

C. Memory Organization

According to derivation (1), ADD operation for a larger HT needs the transform results of four sub-blocks. In this design, HT8x8 adopts 16-pixel parallelism and processes a 16x16 block row by row. We employ a two port SRAM to store the C8. Being accessed with a dedicated order, they are used to calculate the larger HTs in a pipelining manner. Word size of SRAM is 240 bits (16-pel, 15 bits per pel). Two 32x32 blocks are stored with 128 depth. It is written and read in a ping-pong way. Fig. 6 illustrates the organization and access pattern. The data needs to read every eight and 16 addresses to perform the



Technology	E-Shuttle 65nm CMOS
I/O pads	162
Die area	8.82mm^2
Core area	3.29mm^2
Density	81%
Logic Gate Count	1183k (2-input NAND)
SRAM	19.2KB/5.1KB*
Max. resolution	7680x4320 30fps (10bits/pel)
Max. pixel rate	995Mpixels/s

Measured core power

198.6mW for 4320p30 @1.20v, 188MHz
48.3mW for 2160p60 @0.70v, 95MHz
25.9mW for 2160p30 @0.70v, 48MHz
6.3mW for 1080p30 @0.70v, 12MHz

*: 5.1KB SRAM for chip test

Fig. 7. Chip specification and micrograph.

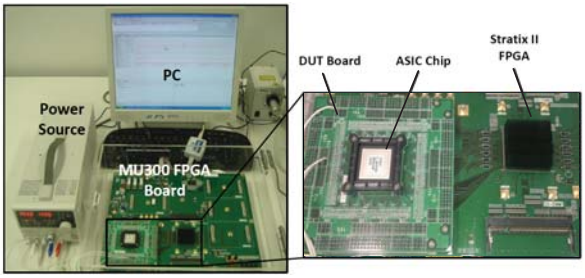


Fig. 8. Photos of verification system.

ADD operations for HT16x16 and 32xx32. Thus, we classify a 32x32 block into 8 groups (G32) by every eight addresses and access the data of groups as the ascending order. By doing so, the HAD16 and HAD32 processing are fully pipelined.

V. Implementation Result

To verify the proposed architecture, it is implemented with 65nm 1P12M LVT CMOS. Fig. 7 shows the die photo and specification of the chip. 1183K logic gates, 24.3KB SRAM (including 5.1KB for test) and a PLL are integrated into a 3.29mm^2 core. The chip is verified by FPGA based evaluation system, as shown in Fig. 8. At 1.2V supply and 188MHz frequency, the design achieves 7680x4320 30fps real-time encoding. The corresponding power dissipation is 198.6mW with 199.4pJ/pixel energy efficiency. At 0.7v, the 1080p30 processing dissipates only 6.3mW when running at 12MHz. The best energy efficiency is 97.0pJ/pixel as measured at 0.7v and 95MHz. Fig. 9 shows the RD performance curves of the design and referenced software HM10.0 for 2160p video "Nebuta". With much lower complexity, our design delivers similar coding efficiency with referenced software.

The comparison with the state-of-art designs on specification and performance are concluded in Fig. 10. With algorithm and architecture co-optimization, this design reaches 995Mpixels/s high throughput for 7680x4320 30fps video, at least 4.7 times faster than previous designs. Despite high complexity in HEVC, our chip achieves much better power efficiency (0.2nJ/pixel) than previous works in H.264, at least 56% improvement, even considering the technique scaling.

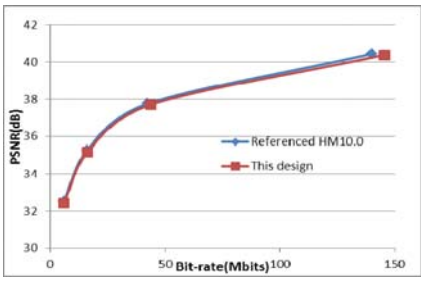


Fig. 9. Rate-distortion curves of referenced HM10.0 and proposed design.

	This Work	TVLSI'10[2]	ISSCC'09[3]/ICME'09[4]
Max. Resolution	7680x4320@30fps	1920x1080@30fps	4096x2160@24fps
Max. Throughput	995Mpixels/s	62.2Mpixels/s	212Mpixels/s
Standard	HEVC(H.265)	H.264	H.264
Transform	HT8x8/16x16/32x32	HT4x4	HT4x4/8x8
Algorithm	8/2-tap & 5T12S	6/2-tap & 9T9Sx2	2-tap & 9T49S
Technology/Supply	65nm/1.2V	0.18um/1.2V	90nm/1.2V
Logic gates/SRAM	1183K/19.2KB	321K/9.72KB	448K/-
Cycles/pixel[1)]	0.19	2.46	1.32
FME power	198.6mW	374mW	135.02mW[2)]
Norm. FME power[3)]	198.6mW	135mW	97.49mW
Power efficiency	0.2nJ/pixel	2.17nJ/pixel	0.46nJ/pixel

1): The speed normalized to the processing cycles for each pixel.
2): FME power calculated from encoder core power and FME area proportion.
3): FME power normalized to 65nm/1.2V. (P65nm/1.2V = P90nm/1.2V /1.385 = P0.18μm/1.2V / 2.77)

Fig. 10. Design comparison with the state-of-art designs.

VI. Conclusion

In this paper, we propose an HEVC FME architecture. A BQA scheme, together with a 5T12S search pattern is proposed to reduce the complexity. ES-HAD is proposed to improve coding quality of FME and determine transform size without RQT. Furthermore, a data reuse method of ES-HAD is applied to reduce hardware cost. The implemented chip achieves 7680x4320 30fps real-time encoding, at 1.2V supply and 188MHz frequency. Despite high complexity in HEVC, the design achieves at least 56% improvement on power efficiency (0.2nJ/pixel) than previous works in H.264.

References

[1] Joint Collaborative Team on Video Coding, "High Efficiency Video Coding (HEVC) text specification draft 10," JVT-G050, Jan. 2013.

[2] C.-Y. Kao, C.-L. Wu, Y.-L. Lin, "A High-Performance Three-Engine Architecture for H.264/AVC Fractional Motion Estimation", IEEE Trans. VLSI Syst., vol.18, No. 4, pp. 662-666, April, 2010.

[3] L.-F. Ding, W.-Y. Chen, P.-K. Tsung, T.-D. Chuang, H.-K. Chiu, Y.-H.Chen, P.-H. Hsiao, S.-Y. Chien, T.-C. Chen, P.-C. Lin, C.-Y. Chang, W.-L. Chen, and L.-G. Chen, "A 212 MPixels/s 4096 x2160p multiview video encoder chip for 3D/quad HDTV applications," in IEEE ISSCC Dig. Tech. Papers, Feb, 2009.

[4] P.-K. Tsung, W.-Y. Chen, L.-F. Ding, C.-Y. Tsai, T.-D. Chuang, and L.-G. Chen, "Single-iteration full-search fractional motion estimation for quad full HD H.264/AVC encoding," in Proc. ICME, 2009, pp. 9–12.

[5] T.-C. Chen, Y.-H. Chen, C.-Y. Tsai, S.-F. Tsai, S.-Y. Chien and L.-G. Chen, "2.8 to 67.2 mW low-power and power-aware H.264 encoder for mobile applications," in VLSI Circuits Symp. Dig., 2007, pp. 222–223.