



High-Throughput Power-Efficient VLSI Architecture of Fractional Motion Estimation

P. Rahul Reddy¹

Associate Professor, Dept of ECE, Swami Ramananda Tirtha Institute of Science & Technology, Nalgonda, India¹

ABSTRACT: The next-generation video coding standard of High-Efficiency Video coding (HEVC) is particularly economical for coding high-resolution video like 8K-ultra-high-definition (UHD) video. Fractional motion estimation in HEVC presents a major challenge in clock latency and area cost because it consumes quite 40 take advantage of the overall encoding time and therefore results in high computational quality. With aims at supporting 8K-UHD video applications, an efficient interpolation filter VLSI architecture for HEVC is proposed during this paper. Firstly, a replacement interpolation filter algorithm supported the 8-pixel interpolation unit is planned during this paper. It will save 19.7 business data processing time on the average with acceptable coding quality degradation. supported the planned algorithmic program, an efficient interpolation filter VLSI architecture, composed of a reused data path of interpolation, an efficient memory organization, and a reconfigurable pipeline interpolation filter engine, is given to scale back the implement hardware area and achieve high throughput.

KEYWORDS: HEVC, Interpolation filter, VLSI, fractional motion estimation (FME).

I. INTRODUCTION

In multimedia system, there are several video coding standards such as, H.264/AVC [4], VC-1 [5], they are the source coding technology basis for digital multimedia applications. Despite of the emerging HEVC standard [6], H.264/AVC is the most mature video coding standard [4] [9]. China Audio and Video Coding Standard (AVS) is a new standard targeted for video and audio coding [7]. Its video part (AVS-P2) had been formally accepted as the Chinese national standard in 2006 [7]. Similar with MPEG-2, MPEG-4 and H. 264/AVC, AVS-P2 adopts block-based hybrid video coding framework. AVS achieves equivalent coding performance with H.264/AVC. There are different coding tools and features in different standards. However, the crucial technologies they employed are very similar with coincident framework. These similar standards are MPEG-like video standards. Ultra-high definition (UHD) video includes 4K (3840×2160) and 8K (7680×4320, also known as super Hi-vision) formats. By delivering 4 and 16 times of pixels per frame compared to today's high definition (1920×1080), UHD videos offers remarkably enhanced visual experience and provides rich cues for stereoscopic feeling such as visual field angle, linear perspective, and texture gradient. The UHD is currently being promoted in the next-generation standard of digital television. To store and transmit the huge volume of UHD video data, efficient and real-time compression is essential. The latest video coding standards such as H.264/AVC and H.265/HEVC provide excellent compression ratio. However, the key compression algorithms, such as intra prediction and fractional motion estimation (FME), involve high computational complexity. Moreover, due to data dependency and the algorithm process, it is limited to apply the pipelining and parallel processing techniques in hardware design. The existing works proposed many architecture designs of intra prediction and FME, mainly targeting for 1080p HD or lower resolution videos. We cannot realize the UHD-throughput design by simply increasing times of previous architectures, since it leads to incredibly large hardware cost and high power for a chip.

Intra prediction, which uses neighboring pixel values to predict the currently coding block, explores spatial redundancy of the video. Fig. 1 shows the H.264/AVC intra-frame encoding flow. Firstly, a prediction generator (PG) unit refers to reconstructed pixels of neighboring block to generate predictive pixels for each mode. The residues are then generated and transformed into coefficients. Based on a given cost function, the costs are calculated to perform mode decision. The quantization unit processes the coefficients of the best mode and outputs the results to both the inverse quantization unit and the entropy coding unit. The inverse quantization and inverse transform units translate quantized values back to residuals, which are used to reconstruct pixels for PG operation of the next block. The entropy coding unit encodes quantized values and mode information for bit-stream output.

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

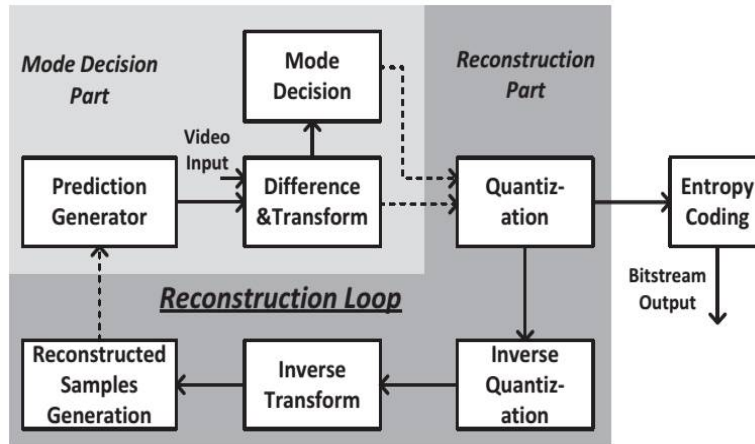


Fig.1. H.264/AVC intra-frame encoding flow

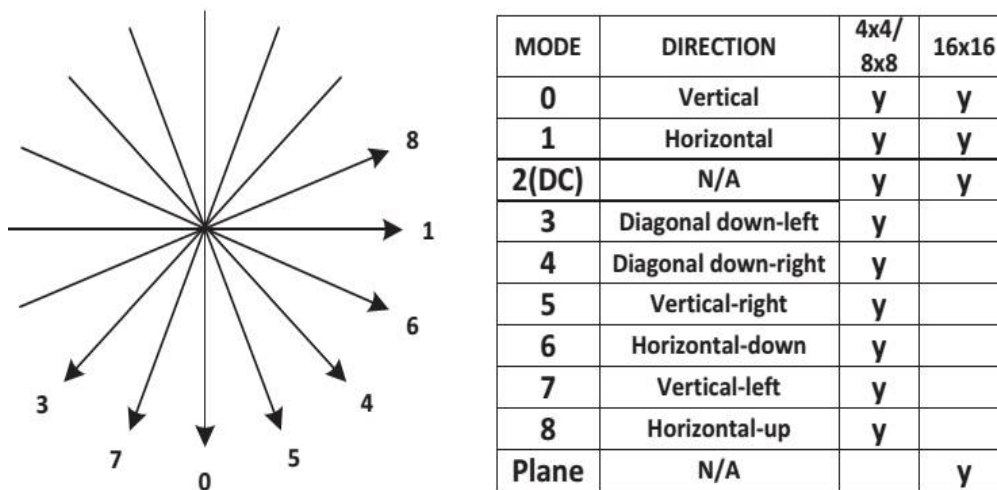


Fig.2. the H.264 intra prediction modes

A. Prediction Mode

The prediction modes for luma component are categorized into three sizes. As shown in Fig. 2, both 4x4 and 8x8 predictions contain eight directional and a DC modes, while 16x16 prediction has two directional (vertical and horizontal), a DC and a plane modes. The 16x16 pixels in an MB can be encoded as 16 predicted blocks with the 4x4 mode, or four predicted blocks with the 8x8 mode, or just a predicted block with the 16x16 mode. In addition, four 8x8 Chroma modes, which are similar to 16x16 luma modes, are adopted to predict the two Cb and Cr 8x8 blocks within an MB.



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

B. Cost Function

The optimal cost function for coding efficiency is the rate distortion optimization (RDO), which brings high computational complexity. In the hardware design that targets at real-time coding for high resolution videos, RDO leads to large hardware cost and power dissipation. For lower complexity, the sum of absolute transform difference (SATD) is commonly used in conventional designs.

The cost of each mode is estimated with transformed coefficients. The computational formula is defined as follows:

$$\text{Cost} = \text{SATD} + \lambda(\text{QP}) \cdot R \dots \dots \dots (1)$$

$$\text{SATD} = \sum \sum |T(\text{Cur} - \text{Pre})| \dots \dots \dots (2)$$

In (1), R is set to 0 for the most probable mode and to 4 for the other modes. The value of λ is a function of the quantization parameter (QP). In (2), T(x) can be either a Hadamard transform (HT) or an integer discrete cosine transforms (DCT). Cur and Pre denote the original and predictive pixels. In addition, the sum of absolute difference (SAD) cost function is also suitable for hardware design. It does not need to perform the transform operation, thus introduces the least complexity.

II. RELATED WORK

There have been many previous works focusing on designing efficient architecture for HEVC MC Interpolations. Huang proposed a high-throughput interpolation filter architecture with a prediction unit (PU)-adaptive filtering flow and a unified filter combining the eight-tap luma and four-tap chroma filters [11]. But its hardware area is larger than the hardware cost proposed in this paper.

In [12], a dedicated hardware accelerator for interpolation was presented. Although it could read 8 input samples and produce 64 output samples at each clock cycle, its area cost was huge. An efficient VLSI design which is composed of a reconfigurable filter, an optimized pipeline engine organization, and a filter reuse scheme for HEVC interpolation was proposed in [13]. This hardware is slower than the architecture proposed in this paper because it has restricted reconfigurability for filter data paths. In [14], a simplified fractional motion estimation (FME) architecture for field-programmable gate arrays (FPGAs) is presented that processes only 8×8 -sized blocks at the cost of a bit rate increase of 13 %. In [15], reconfigurable acceleration engines were developed in the interpolation filter hardware architecture to adapt to different filter types. In [16], a low-energy HEVC sub-pixel interpolation hardware for all PU sizes was proposed and Hcub multiplierless constant multiplication algorithm was used. To overcome the obstacles of the previous work, we proposed a fast interpolation filter algorithm and the corresponding hardware architecture in [18], which can save the encoding time and reduce the computational complexity of fractional motion estimation in HEVC.

III. PROPOSED ALGORITHM

Fractional motion estimation performs a half-pixel refinement about the integer search positions, and then a quarter-pixel one is performed around the best half pixel position. In the interpolation algorithm, it is known that the quarter-pixel interpolation processor needs to filter the results of the half-pixel horizontal interpolation in a vertical direction. If carrying out the interpolation process of a 64×64 CU, $2 \times (64 + 1) \times (64 + 8) \times (8 + 6) = 131,040$ bits RAM is required in total. The area cost will be huge for hardware implementation. In our design, a reused three-level architecture is proposed for half-pixel and quarter-pixel interpolations. With this structure, we would not need to store the intermediate results and thus can reduce the area cost for about 131,040 bits RAM. Fig.3 shows the data path of the interpolation processor. There are three horizontal filters (H_F1/4, H_F2/4, H_F3/4 in level 1) and eight vertical filters (V_F1/4, V_F2/4, V_F3/4 in level 2 and level 3) in the proposed three-level reused architecture.

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

There are three horizontal filters in the first level (level 1). For the half-pixel interpolation as shown in Fig. 3(a), the horizontal filter H_F2/4 is open and the other two are close in the first round. The half-pixel $b_{0,0}$ (as seen in Fig. 1) is calculated by H_F2/4 from the integer pixel $A_{0,0}$ in the horizontal direction. For the quarter-pixel interpolation in the second round as shown in Fig.3 (b), the filtered results of pixels $a_{0,0}$, $b_{0,0}$, and $c_{0,0}$ are calculated by the three horizontal filters in level 1 from the integer position $A_{0,0}$.

The second level (level 2) contains four vertical filters. They work just at the second round of the quarter-pixel interpolation process. The quarter pixels $e_{0,0}$ and $p_{0,0}$ are interpolated by the filters V_F1/4 and V_F3/4, respectively, from the pixel $a_{0,0}$ in the vertical direction. Similarly, the quarter pixels $g_{0,0}$ and $r_{0,0}$ are interpolated, respectively, by the filters V_F1/4 and V_F3/4 from the pixel $c_{0,0}$ in the vertical direction. The last level (level 3) also contains four vertical filters. The difference between the four vertical filters in level 2 and level 3 is that the data inputs of the vertical filters in level 3 are not fixed. The filtered results of the half pixels $h_{0,0}$ and $j_{0,0}$ are calculated by the two vertical filters V_F2/4 from the pixels $A_{0,0}$ and $b_{0,0}$ at the first round of the half-pixel interpolation process. During the second round, the quarter pixels $i_{0,0}$ and $k_{0,0}$ are interpolated by the same two vertical filters from the pixels $a_{0,0}$ and $c_{0,0}$ when the vertical component of the best half MV is not equal to zero.

The interpolated results of quarter pixels $d_{0,0}$ and $n_{0,0}$ are calculated by the other two vertical filters V_F1/4 and V_F3/4 from the integer pixel $A_{0,0}$ when the horizontal component of MV is equal to zero; otherwise, the quarter pixels $f_{0,0}$ and $q_{0,0}$ are interpolated by the same vertical filters V_F1/4 and V_F3/4 from the half pixel $b_{0,0}$. From the above data path of the proposed interpolation filter architecture, it can be seen that all the horizontal and vertical filters in the process of half-pixel interpolation can be reused in the process of quarter-pixel interpolation.

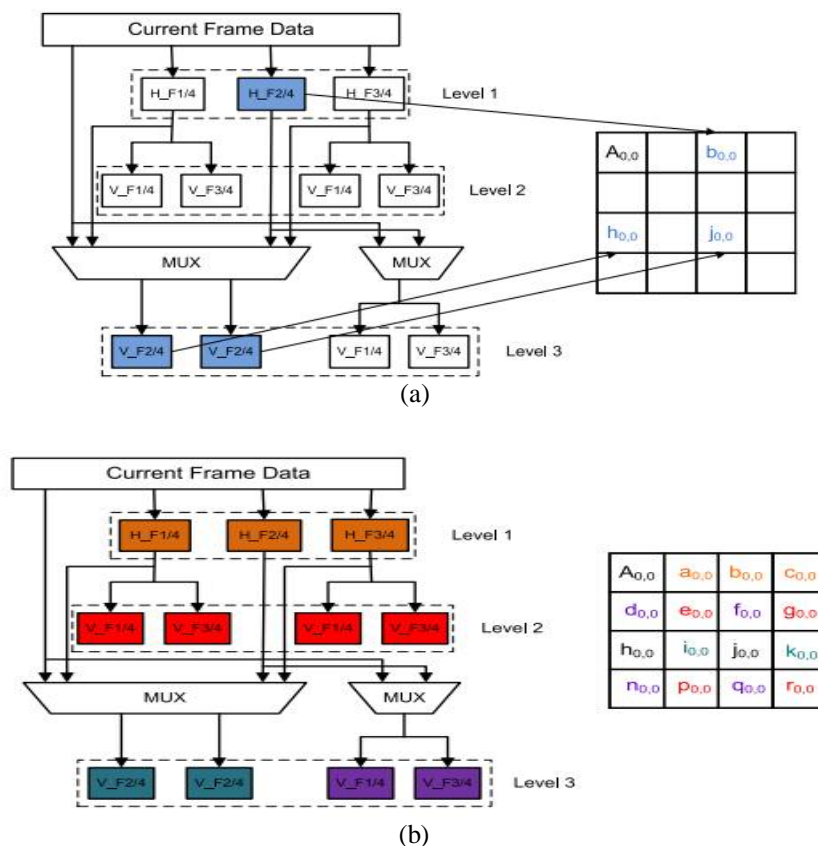


Fig. 3 The reused data path of interpolation filter. (a) First round: half-pixel interpolation.(b) Second round: quarter-pixel interpolation. The reused data path of the interpolation processor



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Vol. 4, Issue 1, January 2016

a) The data path of the first round of interpolation processor for half-pixel interpolation. b) The data path of the second round of interpolation processor for quarter-pixel interpolation. H_F1/4, H_F2/4, and H_F3/4 in level 1 represent three horizontal filters. V_F1/4, V_F2/4, and V_F3/4 in level 2 and level 3 represent eight vertical filters. MUX represents multiplexer.

IV. CONCLUSION

In this paper, high-performance VLSI architecture for luma interpolation in HEVC is proposed and it is implemented with 37.2k gates at an operating frequency of 240 MHz. It can support 8K-UHD (7680 × 4320)@78fps (4:2:0 format) real-time video processing. Our proposed architecture can be reused for half-pixel interpolation and quarter-pixel interpolation, and it diminishes the area cost of RAM with the reused interpolation architecture. Our proposed architecture can accomplish high throughput for real-time encoding of ultra-high-resolution videos with reduced hardware resources and is specifically suitable for 8K-UHD video real-time encoding.

REFERENCES

- [1] Joint Video Team, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)," JVT-G050, May 2003.
- [2] Li-Fu Ding, Wen-Yin Chen, Pei-Kuei Tsung, Tzu-Der Chuang, Pai-Heng Hsiao, Yu-Han Chen, Hsu-Kuang Chiu, Shao-Yi Chien and Liang-Gee Chen, "A 212Mpixels/s 4096x2160p Multiview Video Encoder Chip for 3D/Quad Full HDTV Applications," IEEE Journal of Solid-State Circuits, vol. 45, no. 1, pp. 46-58, 2010.
- [3] J-R Ohm, GJ Sullivan, H Schwarz, TK Tan, T Wiegand, Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC). IEEE Trans. Circuits Syst. Video Technol. 22(12), 1669–1684 (2012).
- [4] ITU-T (2005). Recommendation and International Standard of Joint Video Specification. ITU-T Rec. H.264/ ISO/ IEC AVC, Mar., 14496-10.
- [5] SMPTE 421M. VC-1 Compressed Video Bit stream Format and Decoding Process. http://www.smpte.org/smpite_store/standards/pdf/s421m.pdf.
- [6] Documents of the first meeting of the Joint Collaborative Team on Video Coding (JCT-VC)- Dresden, Germany. (2010). 15-23 April, ITU-T. 23 April. Retrieved 21 May.
- [7] Information technology- Advanced coding of audio and video- Part 2: Video. (2005). AVS Standard Draft. [8] Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the H. 264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol Jul., 13, 560-576.
- [8]. S. Oktem and I. Hamzaoglu, in Proc.10th Euromicro Conference on Digital System Design. An efficient hardware architecture for quarter-pixel accurate H.264 motion estimation (IEEE, Luebeck, Germany, 2007)
- [9] <http://en.wikipedia.org/wiki/x264>.
- [10]. D. Zhou and P. Liu, in Proc. IEEE International Symposium on Circuits and Systems. A hardware-efficient dual-standard VLSI architecture for MC interpolation in AVS and H.264 (IEEE, New Orleans, Louisiana, 2007).
- [11]. Chao-Tsung Huang, Chiraag Juvekar, Mehul Tikekar, Anantha P. Chandrakasan, in Proc. IEEE Conference on Visual Communications and Image Processing (VCIP). HEVC interpolation filter architecture for quad full HD decoding (IEEE, Kuching, Sarawak, 2013)
- [12]. G. Pastuszak, M. Trochimiuk, in Proc. 16th Euromicro Conference on Digital System Design. Architecture design and efficiency evaluation for the high throughput interpolation in the HEVC encoder (IEEE, Santander, Spain, 2013)
- [13]. Guo Z, Zhou D, Guto S, in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). An optimized MC interpolation architecture for HEVC (IEEE, Kyoto, Japan, 2012)
- [14]. V. Afonso, H. Maich, L. Agostini, and D. Franco, in Proc. IEEE Lat. Amer. Symp. Circuits Syst. (LASCAS). Low cost and high throughput FME interpolation for the HEVC emerging video coding standard (IEEE, Cusco, Peru, 2013)
- [15]. CM Cláudio, M Shafique, S Bampi, J Henkel, A reconfigurable hardware architecture for fractional pixel interpolation in high efficiency video coding. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 34(2), 238–251 (2015)
- [16]. E. Kalali, I. Hamzaoglu, in Proc. IEEE International Conference on Image Processing (ICIP). A low energy HEVC sub-pixel interpolation hardware (IEEE, Paris, French, 2014)

BIOGRAPHY



P. Rahul Reddy attained his B.Tech in Electronics & Communication Engineering and M.Tech in the stream of Embedded Systems from JNTU, Hyderabad. He is having teaching experience of more than 5 years in various Under Graduate and Post Graduate course. He has guided lots of students in various Under Graduate and Post Graduate Research Projects. At present, he is working as Associate Professor, Department of ECE in Swami Ramananda Tirtha Institute of Science & Technology, Nalgonda, India.