FULLY UTILIZED AND REUSABLE ARCHITECTURE FOR FRACTIONAL MOTION ESTIMATION OF H.264/AVC

Tung-Chien Chen, Yu-Wen Huang, and Liang-Gee Chen

DSP/IC Design Lab., Graduate Institute of Electronics Engineering and
Department of Electrical Engineering, National Taiwan University
{djchen, yuwen, lgchen}@video.ee.ntu.edu.tw

ABSTRACT

In this paper, we contributed a new VLSI architecture for fractional motion estimation of H.264/AVC. Seven inter-related loops extracted from complex procedure are analyzed and two decomposing techniques are proposed to parallelize the algorithm for hardware with regular schedule and full utilization. The proposed architecture, also characterized by reusable feature, can support situations in different specification, multiple standards, fast algorithm and some cost considerations. H.264/AVC baseline profile Level 3 with complete Lagrangian mode decision can be realized with 290K gates at operating frequency of 100MHz. It is an useful Intellectual Property (IP) design for platform based multimedia system.

1. INTRODUCTION

The H.264/ AVC video compression standard[1], jointly developed by ITU-T and ISO/IEC, provides at least 2x compression improvement[2] and substantial perceptual quality enhancement over all previous standards but significantly increases the computation complexity. According to experimental results from JM7.3[3] in baseline profile Level 2 with CIF format, 5 reference frames,and $\pm$16SR, the encoding procedure requires 80G instructions per second which is dominated (99%) by the inter prediction for new techniques of variable block sizes and multiple reference frames with Lagrangian mode decision. Therefore hardware acceleration is a must. Several fast algorithms and hardware architecture are proposed for integer motion estimation (IME) to meet real-time requirement, but for fractional motion estimation (FME) which occupies 45% of the run-time in inter prediction and upgrades rate-distortion efficiency by 4+ dB in PSNR. Obviously, encoding procedure of FME unlike previous standards must be MB pipelined with IME and processes by a dedicated module.

The main difficulty for hardware implementation of FME is the complex mode decision flow. Because Lagrangian mode decision is done between costs of 41 sub-blocks in every reference frame with quarter precision, the FME flow contains seven inter-correlative loops with operations of interpolation, residue generation, and hadmard transform. According to our analysis, we decompose the flow by techniques of 4x4 block decomposition and vertical integration to map and parallelize several loops in hardware with features of regular schedule, full utilization, and reusability. The compliant architecture for one reference frame is proposed with 36-times parallelism, and we can support baseline profile level 3 with four such FME module at 100 MHz operation frequency. The rest of this paper is organized as follows. In Section 2, we analysis and decompose the loops to sketch the FME module. In Section 3, the compliant architecture including MC function is proposed. The proposed fast algorithm and low cost considerations discussed in Section 4 manifest the reusability of our design. Afterwards, Section 5 shows the implementation results and Section 6 gives a conclusion.

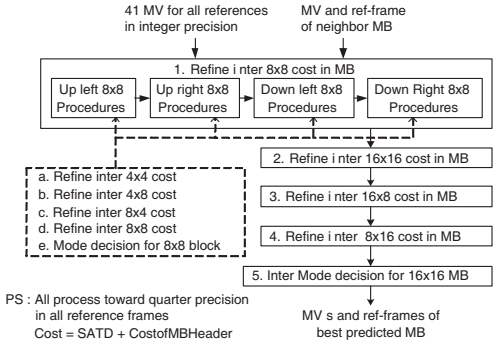


Fig. 1. Flow of Lagrangian mode decision for FME in H.264

2. INTER PREDICTION PROCEDURE OF FME IN H.264

Figure.1 shows the inter prediction flow of a 16x16 macro-block (MB) for fractional motion estimation (FME) in H.264. Unlike traditional mode decision algorithm, the Lagrangian mode decision is done after costs of all kinds of block sizes and reference frames are computed in quarter precision, which improves the coding performance by 0.5$\sim$1.5 dB in PSNR. The half-pixel ME refinement is performed around the best integer search positions of 41 blocks at all reference frames, and quarter-pixel ME, as well, is performed around the best half search positions. The mode decision considers not only the sum of absolute transformed difference (SATD) but also the exact cost of required bits of MB header including MV's, reference frames, and prediction modes. The latter results in inevitable sequential processing and conflicts the MB pipeline schedule due to data dependencies between neighboring MB's and sub-blocks. The complex sequential procedure with hardware-unfriendly algorithm is the main difficulty of hardware implementation with real time requirement. In the following, we will analyze the procedure of FME and decompose it to map in efficient hardware with regular procedure. The hardware-oriented algorithm that removes data dependencies conflicting MB pipeline is also applied with no degradation in video quality.

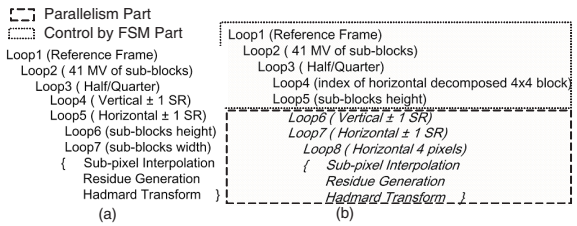


Fig. 2. (a) Original seven loops of FME procedure. (b) Decomposed loops after 4x4 Block decomposition and Vertical integration

2.1. Procedure decomposition and analysis

For simplification, we decompose the whole procedure of FME into seven iteration loops as Fig.2(a). The first two loops are sub-blocks from different block types and reference frames that are selected to form a predicted MB. The third loop sequentially processes each sub-block in half and then in quarter precision. The following two loops contain nine candidates within horizontal and vertical ± 1 search range(SR), and the last two are iterations of pixels in each sub-block. The core procedures inside the loops include interpolation, residue generation, and Hadamard transform. Different from residue generation, interpolation is 6-tap FIR in both horizontal and vertical direction and Hadamard transform is a 2-D 4x4 operation. We must parallelize some loops in hardware to meet the real-time requirement and integrate three core procedures to get full hardware utilization.

We start from one reference frame and ignore the first loop. The second loop of sub-block types includes 41 MV pointing to different positions, and ultra random access of search window(SW) memory is required if this loop is parallelized. The third loop cannot be parallelized for sequential manner of half and quarter refinement procedure. Opposite to the previous loops, the candidate loops are suitable for parallelism because interpolated pixels can be reused for adjacent candidates of each sub-block, and redundant computations and cycles will be saved. Parallelization within the last two loops is involved with the coordination and hardware utilization of three core procedures. However, the iteration here depends on block size and ranges from 4 to 16, which make the cooperation a tough job. Two techniques that consider regularity, reusability, and utilization of each process unit are proposed in the following.

2.1.1. 4x4 Block decomposition

The 4x4 block is the smallest element for sub-blocks and SATD transform procedure in H.264. Every sub-block in MB can be decomposed to several 4x4 element blocks with the same MV's. Therefore, we can concentrate on designing a processing unit (PU) of 4x4 block and reuse it in all block types. As Fig.3(a), 4x8 sub-block is decomposed into two 4x4 element block, and the SATD of each element block is accumulated to get the final cost. Actually, we arrange nine 4x4 block PU's to manage the nine candidates around the refinement center simultaneously, and each PU has 4-times parallelism in adjacent horizontal pixels for residue generation and decomposed Hadamard transform[4]. The decomposed flow is shown in Fig.2(b) with 36-times parallelization for one reference frame. The interpolation procedure dominates the operation time that is 10 cycles for one 4x4 element block.

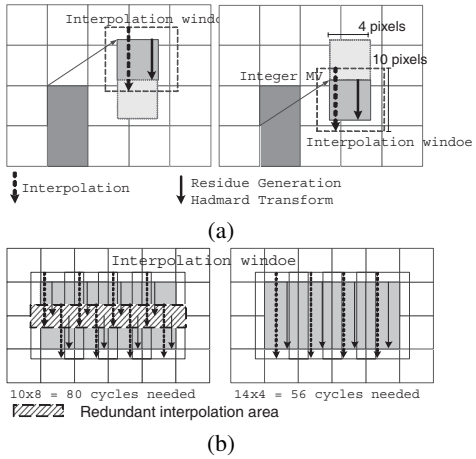


Fig. 3. (a)4x4 block decomposition of 4x8 sub-block. (b)Vertical integration of 16x8 sub-block.

Table 1. Hardware utilization of different sub-blocks.

block type	block number	cycle/ block	PE utilization	Interpolation utilization
Inter 16x16	1	22x4	73%	100%
Inter 16x8	2	14x4	62%	100%
Inter 8x16	2	22x2	73%	100%
Inter 8x8	4	14x2	57%	100%
Inter 8x4	8	10x2	40%	100%
Inter 4x8	8	14x1	57%	100%
Inter 4x4	16	10x1	40%	100%
PU average utilization : 64% Interpolation utilization : 100%				
Total operation time for FME : 824 x 2 = 1648 cycle/MB				

2.1.2. Vertical integration

After 4x4 block decomposition, redundant interpolating operations appear in overlapped area of adjacent interpolation window. The operation time dominated by interpolation can be reduced if vertical adjacent 4x4 blocks are integrated and overlap interpolated data are reused by the following element block. As Fig.3(b) shows, 30% of the cycles are saved, and PE (subtract and absolute) utilization is improved from 40% to 57% for 16x8 sub-block. Indeed, the improvement varies with the height of each sub-block (Tab.1), and the 27% of the cycles needed for FME is reduced with 64% and 100% utilization for PE and interpolation, respectively.

2.2. Algorithm Modification

Lagrangian mode decision takes motion vector predictors (MVP) into account, which improves coding performance significantly but causes data dependencies between neighboring MB's and sub-

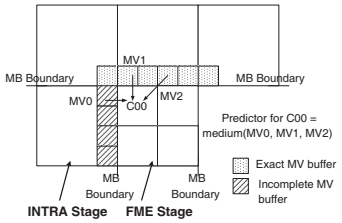


Fig. 4. Incomplete MV predictors used in FME stage.

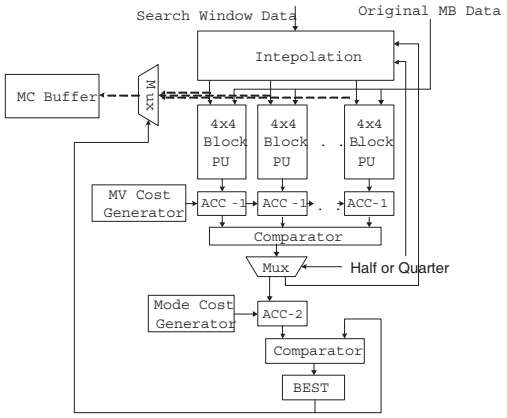


Fig. 5. Block diagram of FME hardware.

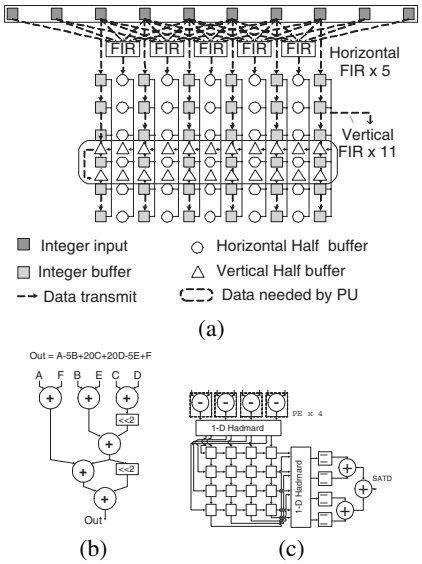


Fig. 6. Architecture of (a)Interpolation Unit (b)6-tap 1D FIR (c)4x4 block PU

blocks and prevents parallel processing and MB pipelining. Fortunately, its computational complexity is not as high as Integer ME. We can carefully map the sequential flow to hardware with parallelism model mentioned previously. However, the MB pipeline problem still exits. For a MB in FME, its left MB is processed intra prediction in next MB pipeline, and the MV's from left MB are incomplete (before intra/inter mode selection). Therefore, exact MV's in left MB are replaced by incomplete ones. As Fig.4 shows, exact MV0 is not available in FME stage and is replaced by its incomplete version to estimate the MVP of 8x8 block, C00. By simulating in software, the compression performance is the same after such modified algorithm applied. For following entropy coding after entire mode decision, the exact MVP's must be calculated in the sequential order defined by standard.

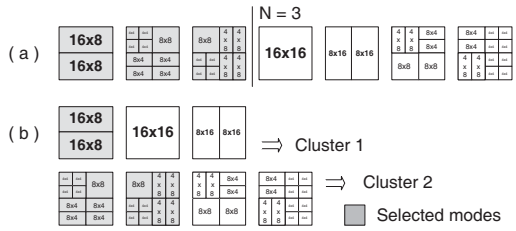


Fig. 7. (a)fast algorithm of AMPD. (b)fast algorithm of AMPD2.

3. ARCHITECTURE

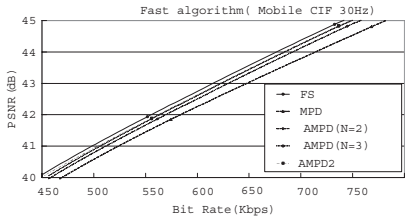
In this section, we will describe the proposed architecture for fractional ME module with the procedure characterized by regular flow and efficient hardware utilization, which we mentioned before. Fig.5 shows the block diagram. The 4x4block PU that has four times parallelization of horizontal adjacent pixels is responsible for residue generation and Hadamard transform. It processes 4x4 element blocks decomposed from sub-block in sequential order. The architecture of each PU is shown in Fig.6(c), four processing elements (PE's) and decomposed Hadamard transform contained two 1-D transform and a transpose register array[4] can successively process four pixels in each cycle without any latency. There are nine 4x4block PU's processing nine candidates around the refinement center simultaneously. Four pixels in original MB are broadcasted to every PU at each time and the sub-pixels demanded for PU's are provided by interpolation unit. The operations of 2-D FIR are decomposed to two 1-D FIR's with an interpolation buffer. As Fig.6(a) , ten (3+4+3) adjacent integer pixels in SW memory must be able to access arbitrarily for 5 half pixels filtered horizontally. These 5 half pixels with six integer ones are latched and shifted in interpolation buffer, and the rest 11 pixels are fabricated by filtering corresponding vertical line buffer. The dotted rectangle in the middle of Fig.6(a) stands for all pixels needed for 9 PU's each cycle. As for quarter refinement, other bilinear filters inputted from the dotted rectangle are responsible for quarter-pixels generation.

The circuit in bottom of comparator-1 in Fig.5 is responsible for sequential procedure in mode decision. ACC-1 takes care of prediction cost of each candidate by accumulating SATD of each 4x4 element block and corresponding MV cost. ACC-2 takes care of costs of 8x8 and 16x16 block by accumulating the best predicted cost of each sub-block in quarter prediction and related mode cost. The information of best MB candidate is latched in BEST buffer. The whole procedure of FME is finished after inter mode decision with all cost of 41 sub-blocks in quarter precision. As for motion compensation (MC) which is allocated in the same MB pipeline stage for resource sharing(Interpolation unit) and bandwidth reduction(SW memory), the predicted pixels are selected from interpolation unit according to the best predicted information in BEST buffer. The compensated MB is buffered, and will be transmit to next stage for following coding procedure.

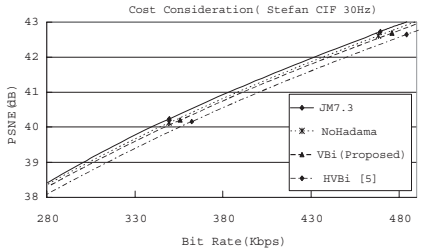
4. REUSABILITY

4.1. Compatible Fast algorithm

The same as IME, fast algorithm is needed for FME in variant situations. Fast algorithm called Mode Pre-Decision(MPD), to decide inter mode in IME phase like traditional standards, can re-



(a)



(b)

Fig. 8. (a)RD curve of fast algorithms. (b)RD curve of cost considerations.

duce much computation complexity and is compatible with our FME module by modifying the control part in Fig.2(b). However, it will seriously degrade the rate-distortion efficiency up to 1 dB in PSNR. To maintain the video quality but also reduce computation, we modify MPD and propose Advanced Mode Pre-Decision Algorithm (AMPD). The main concept of AMPD is that we choose not one but a set of candidate modes during IME phase to process the following FME procedure. First, all 41 sub-blocks are merged into seven MB's and sorted according to the cost generated in integer precision. As Fig.7(a), for example, 16x8 mode has the lowest cost and 8x8 mode with sub-block 4x4,8x8,8x4,8x4 is the next one. Afterwards, N (N=1∼7) of best candidates are picked for FME. If the picked modes have more than one 8x8 modes, each 8x8 position must be sequentially process among the picked sub-mode, like step 1 in Fig.1. As simulation,PSNR improvement with increasing N saturates when N is 3. For further simplifying the AMPD, we classified seven merged MB's as Fig.7(b). One best candidate is selected form the cluster A, and two from cluster B. The new algorithm, called AMPD2, perform as well as AMPA, and has simpler control and regular flow. In our target encoder design which supports full search algorithm, the fast algorithm mode will be turn on if FME dominate the whole system for small SR in IME or in battery low condition. The simulation results are in Fig.8(a). The quality for both algorithms is at most 0.2dB different from FS with less than half computation complexity.

4.2. Reusability for different specifications, multi-standards and low cost considerations

Because of 4x4 based decomposition through all procedures, we can easily support other standards and higher specifications. For example, to be compliant with baseline profile in H.264 Level 3 that supports four reference frames, four FME modules are parallel connected and each of which is responsible for one reference frame. Other standards like MPEG-4 is also compatible but the corresponding change in interpolation unit is needed. As for low cost considerations, we can replaced hardware of 2-D FIR with simpler interpolation scheme, the mismatch will reduce the coding performance. Compared with HVBi, bilinear filter in both horizontal and vertical direction, adopted in [5], our method of Vertical Bilinear Horizontal FIR(VBi) improves quality by 0.3 dB for employing the feature of the higher motion and resolution horizontally in most sequences. VBi can save 11 vertical FIR and most interpolation buffer, which reduces area by 15%. Hadamard transform performing simple transform to estimate bit-rate influenced by DCT can be turn-off if a low cost hardware are needed. 40% area is saved if Hadamard transform is discarded. The RD-curve influenced by the low cost considerations are shown in Fig.8(b).

Table 2. Implementation result of proposed FME module.

	Gate Count
Interpolation Unit	23872
MVCost Core	6477
PUx9	34839
Mode Decision	2174
Control	1538
InOutBuffer	10472
Total	79372

5. IMPLEMENTATION

An architecture of FME with function of complete Lagrangian mode decision and MC for H.264 has been proposed and synthesized in UMC 0.18u technology. The gate count of each portion is listed in Tab.2. It can process 49K MB/Sec in 100 MHz and is sufficient to support SDTV format in 30Hz for one reference frame. It is a cost efficient accelerator solution with full hardware utilization for platform-based MB pipeline video coding design.

6. CONCLUSION

This paper presents a VLSI architecture design for fractional motion estimation of H.264/AVC. We analyze the Lagrangian mode decision loops and provide decomposing methodologies to obtain the optimized projection in hardware implementation. The proposed architecture is attractive for regular flow, high utilization and reusable feature and can support baseline profile Level 3 with four such FME module in 100 MHz operation frequency.

7. REFERENCES

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

[2] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, July 2003.

[3] *Joint Video Team Reference Software JM7.3*, http://bs.hhi.de/ suehring/tml/download/, Aug. 2003.

[4] T. C. Wang, Y. W. Huang H. C. Fang, and L. G. Chen, "Parallel 4x4 2D transform and inverse transform architecture for MPEG-4 AVC/H.264," in *Proc. of ISCAS*, 2003.

[5] T. C. Wang, Y. W. Huang H. C. Fang, and L. G. Chen, "Performance analysis of hardware oriented algorithm modifications in H.264," in *Proc. of ICASSP*, 2003.