

Complexity Reduction for SOPC-based H.264/AVC Coder via Sum of Absolute Difference

Ruei-Xi Chen*, Jeffrey Fan**

*Dept. of Computer Science and Information Engineering, St. John's University, Taipei, Taiwan, Email: crx@mail.sju.edu.tw

**Dept. of Electrical & Computer Engineering, Florida International University, Miami, Florida 33174, USA, Email: jeffrey.fan@fiu.edu

Abstract

There exist a number of complex encoding techniques which make H.264 video coding much more efficient, such as the adoption of variable block sizes, multiple reference frames, and the consideration of Rate-Distortion Optimization (RDO). However, these techniques come with a price, i.e. considerable increase of complexity due to the introduction of Motion Estimation (ME) and Mode Decision in the design of H.264. In this paper, we have proposed a cost-effective complexity reducing coding algorithm for removing H.264 ME redundancy in SOPC-based embedded systems. The loosely coupled accelerators for Avalon switch fabric compliant topology reveal that the potential coder design can achieve the advantages of flexibility and performance in circuit design without incurring much of the design risk.

Keywords

SOPC-based embedded system, Motion estimation (ME), variable blocks size, rate-distortion optimization, mode decision.

1. Introduction

Based on hybrid motion compensation and transform coding model, the emerging video coding standard like H.264/AVC [1] is capable of achieving the best video quality under certain bit rate constraints. Three key features are employed to make H.264 more efficient in terms of performance in comparison to older standards such as MPEG-1/2/4, and H.263. The features include the adoption of variable block sizes, multiple reference frames, and the consideration of Rate-Distortion Optimization (RDO) as a non-normative tool within the codec in motion estimation (ME) scheme [2][3][9]. Unfortunately, these new features also pay the penalty of considerable increase in terms of complexity in encoder, mainly with regards to Motion Estimation and Mode decision. This means that it may bring over than 60% of complexity for the whole encoding work. Constructing an efficient ME engine to support variable block sizes and multiple reference frames has become an essential issue that breaks the key bottleneck of the coder design for H.264 compliant circuits. Academic and industrial research and development were carried out for the problem of ME complexity reduction. Novel

algorithmic and architectural solutions that focus on the ME tasks for the implementation of context-aware coprocessors in real-time, low-power embedded systems are proposed in [7]. A FPGA-based complexity-reduced three dimensional motion estimation (3D-ME) hardware recursive architecture which significantly reduces the number of search candidates and hence the gate count suitable for DSP processor SOC designs is presented in [8]. A Sum of Absolute Difference (SAD) reuse based three level pyramid hierarchical ME algorithm for high performance hardware architecture for FPGA implementation is proposed in [4][5]. In this paper, we focus on the demonstration of a potential complexity reducing SAD reuse principle for H.264/AVC video coder. The SAD reuse principle is a cost-effective approach for removing ME redundancy and capable of realization with a loosely coupled accelerators topology in SOPC-based embedded systems.

This paper is organized as follows. In Section 2, the features of H.264 video coding standard are introduced. The complexity reducing algorithm is derived in Section 3, followed by an SOPC-based embedded system for complexity reducing principle that is proposed in Section 4. Finally, the conclusion is depicted in Section 5.

2. Key Features of H.264/AVC

H.264 video coding standard is proposed in May 2003 by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG) as the product of a partnership effort, known as the Joint Video Team (JVT). It is also known as MPEG-4 Part 10, or AVC (Advanced Video Coding). Therefore, it is common to refer to the standard as H.264/AVC (or AVC/H.264 or H.264/MPEG-4 AVC or MPEG-4/H.264 AVC) to emphasize the common heritage [10]. Two intent properties are introduced for H.264/AVC. Firstly, it is considered to provide excellent video quality at substantially lower bit rates without increasing the complexity of design. Secondly, it is the consideration of flexibility for a wide variety of applications, such as the design for both low and high bit rates, for low and high resolution video, and the consideration of improving the network efficiency.

There are several new features allow H.264/AVC to code video signals quite efficiently with flexibility [11]. Some of such key features include:

- Variable block size support for motion compensation with lump block sizes down to 4x4, in conjunction with 4x4 level transformations.
- Quarter-sample motion vector accuracy.
- Extended reference frame selection for P frames, among various previously decoded frames.
- De-blocking filter within the motion-compensated prediction loop.
- New context-based adapted entropy coding methods: CAVLC and CABAC.

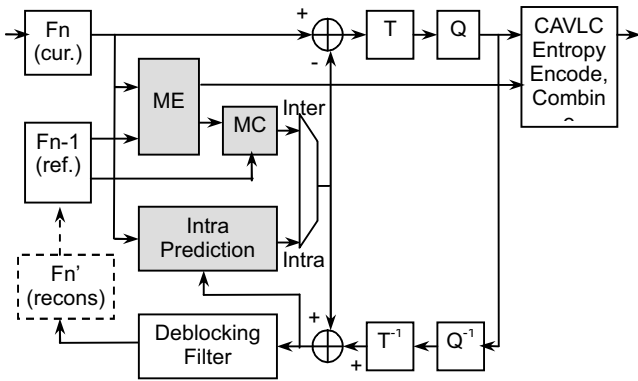


Figure 1 - H.264 Encoder Block Diagram

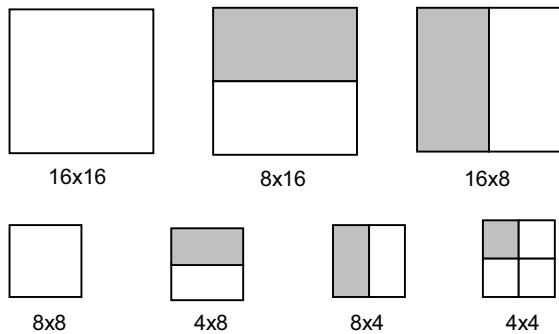


Figure 2 - Seven block sizes is defined in H.264

In H.264 standard, the video coding efficiency is achieved through the combination of a number of encoding techniques. As shown in Figure 1, in addition to the transform coding, quantization, and the entropy coding, the variable block size motion estimation used in the baseline profile of H.264 plays an important key role in achieving better coding results.

However, the increasing encoding complexity caused by the variable block size motion estimation makes it an exciting challenge for realization of the coder. As shown in Figure 2, seven different modes for the variable block size are defined. However, in general they can be divided into two categories: (1) Macroblock partitions consisting of 16x16, 16x8, 8x16 block sizes; (2)

8x8 mode sub-block partitions consisting of 8x8, 8x4, 4x8, 4x4 block sizes, as shown in Figure 2.

Furthermore, other three H.264 modes are introduced in P frame: P-skip, Intra-4x4, Intra-16x16. The P-skip mode means a direct copy of previous frame. The other two modes are intra coding modes to encode the current macroblock.

3. Complexity Reducing Algorithms

The variable block size ME allows a 16x16 macroblock to be partitioned into the other sub-blocks as shown in Figure 2. From all of these sub-blocks, there are as more as 41 motion vectors to be determined in the mode decision process [5]. The selection of efficient algorithms to reduce the computational cost for the variable block size in ME algorithm is needed. An SAD reuse algorithm is presented below.

3.1 ME and mode decision algorithms

Based on an RDO framework to find the optimal motion vector (MV) among all sub-block modes, the reference software for H.264 is presented in [6]. The best selection of the mode can be determined by finding the minimal Lagrangian cost [2][4], we summarized the algorithm as the following:

- (1). The Lagrangian cost for both ME and mode decision is:

$$J_m(d) = SAD_M(d) + \lambda_m \cdot R(d - p), \quad (1)$$

where,

M: a Macroblock mode of size $m \times n$, $(m, n) \in \{(4, 4), (4, 8), (8, 4), (8, 8), (16, 8), (8, 16), (16, 16)\}$.

SAD_M : The sum of absolute difference for a block type M.

λ_m : the Lagrangian multiplier for ME.

$d = (d_x, d_y)$: the current MV being considered.

$p = (p_x, p_y)$: the MV prediction used by H.264 during the coding process.

$R(d-p)$: the bitrate spent for coding MV difference information.

- (2). SAD is computed as:

$$SAD_M(d) = \sum_{x=0, y=0}^{m-1, n-1} |c(x, y) - r(x + d_x, y + d_y)|, \quad (2)$$

where,

$c(\cdot)$ and $r(\cdot)$ represent the video data of current and reference frames, respectively.

3.2 The complexity reducing principles

We derive the SAD Reuse Algorithms below:

(1). The SAD index definition

[Definition] Let (i,j) be a two dimensional index for the block mode of size 4×4 in which a sub-block partition is made from the block mode of size 16×16 in H.264, so that the range of i, j is limited from 0 to 3. And, the SAD of the 4×4 block located at (i,j) position is denoted as $SAD_{4 \times 4}^{(i,j)}(d)$, which is shown in Figure 3, then:

$$SAD_{4 \times 4}^{(i,j)}(d) = \sum_{y=4i}^{4i+3} \left(\sum_{x=4j}^{4j+3} |c(x,y) - r(x+d_x, y+d_y)| \right),$$

$$(0 \leq i < 4, 0 \leq j < 4)$$

(3)

According to the definition, the SAD reuse mechanism can be derived from equation (2) by using the common data of $SAD_{4 \times 4}^{(i,j)}(d)$.

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

(a) The $SAD_{4 \times 4}$ index (i,j)

0,0	4,1	8,2	C,3
0,4	4,4	8,4	C,4
0,8	4,8	8,8	C,8
3,0	3,1	3,2	3,3

(b) The left-top coordinate of $c(x,y)$ in each 4×4 block

Figure 3 - The $SAD_{4 \times 4}$ index and the corresponding coordinate

(2). The SAD reuse algorithm

[Theorem] Determination of the SAD reuse terms: For an H.264 macroblock mode of block size $m \times n$, given a candidate motion vector d that located within the search range $[-R, R]$, there exists a $(m/4) \times (n/4)$ number of corresponding SAD reuse terms which are composed of the block mode with size 4×4 and each of their value is based on the basic computation for 4×4 SAD. The SAD reuse function is therefore can be derived as:

$$SAD_M(d) = SAD_{m \times n}(d) = \sum_{i=0}^{\frac{m}{4}-1} \sum_{j=0}^{\frac{n}{4}-1} (SAD_{4 \times 4}^{(i,j)}(d)),$$

(4)

where $(m/4)$ and $(n/4)$ represent the number of row block for size 4×4 sub-block and the number of column block for the size 4×4 sub-block that can be reused, respectively. So that the total number of SAD reuse terms for a given candidate motion vector can be simply calculated as $(m/4) \times (n/4)$. For example, given a mode of block size 4×8 , the $SAD_{4 \times 8}(d)$ then can be computed with the expression:

$$SAD_M(d) = SAD_{4 \times 8}(d) = \sum_{i=0}^0 \sum_{j=0}^1 (SAD_{4 \times 4}^{(i,j)}(d))$$

$$= SAD_{4 \times 4}^{(0,0)}(d) + SAD_{4 \times 4}^{(0,1)}(d),$$

(5)

4. SOPC-based Complexity Reducing H.264 Coder

There are two types of speedup mechanisms that can be used in SOPC-based embedded system [12]. One is the custom instruction, and another is the custom accelerator. The custom instruction is the type of tightly coupled topology, and the custom accelerator is the type of loosely coupled topology. Due to the outstanding of the efficiency, the custom accelerator topology is the preferred consideration for the implementation of H.264 video coder. As shown in Figure 4, the platform-based reference design with Soft-CPU cores and the standard peripherals construct a fundamental paradigm of the SOPC embedded system. The design accommodates a memory resource consumption property in its nature. Therefore, the use of the external SDRAM is required. Unfortunately, the data bandwidth requirement is another bottleneck for the implementation of most ME algorithms. For smoothing the data pumping rate, the interfacing architecture of a distributed buffer with multiple-ports and a memory controller is introduced, as shown in Figure 4.

To design cost-effective SAD reuse architecture, the search range and the reuse methodology for ME must be considered first. A macroblock with the search range $[-p, p]$ will contain a $(2p+1)^2$ search locations. Hence, the

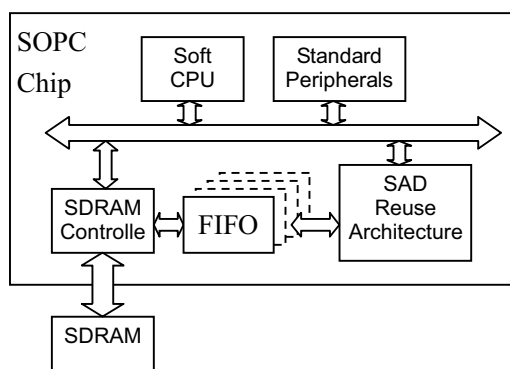


Figure 4 - SOPC-based SAD Reuse Video Coder

basic SAD of block size 4×4 has a $[-4, 4]$ search range, and there are $(2 \times 4 + 1)^2 = 81$ search locations. According to eq. (5), if a row-wised data transfer is used, we can first compute the SAD 1×4 during every data shifted in. Four sets of such a mechanism can be working together in parallel for processing four row data simultaneously. So that we can add the SAD 1×4 together from the four corresponding rows and the results of SAD 4×4 is stored for further use. The processing element (PE) used for such an SAD reuse computing can be designed with the behavior as: SAD 4×4 results computing, the combination of SAD for mode decision, and the motion vector determination, etc.

5. Conclusion

In this paper, we have presented the complexity reducing algorithm with the SAD reuse architecture contributed to the H.264 coder. It has great potential to employ the principle to realize an efficient video encoder on the SOPC-based embedded systems. The cost-effective SOPC architecture design for possible embedded H.264 video coder implementation is proposed. Furthermore, future research may resolve some interesting issues, such as the exploration of hardware/software co-design techniques, and the optimization of the embedded real-time system for H.264/AVC video coder.

6. REFERENCES

- [1] Joint Video Team of ITU-T and ISO/IEC JTC 1, "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC)," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Document JVT-GO50, December 2003.
- [2] H.-Y. C. Tourapis, A.M. Tourapis, "Fast motion estimation within the H.264 codec," Proc. IEEE Int. Conf. Multimedia and Expo, ICME '03, vol. 3, pp. III - 517-20, July 2003
- [3] He Zhihai, S.K. Mitra, "A unified rate-distortion analysis framework for transform coding," IEEE Trans. Circuits and Systems for Video Technology, Vol. 11, Iss. 12, Dec. 2001, pp. 1221-1236.
- [4] H.F. Ates, Y. Altunbasak, "SAD reuse in hierarchical motion estimation for the H.264 encoder," IEEE Int. Conf. Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05), Vol. 2, Mar. 2005, pp. II/905-ii/908.
- [5] S. Yalcin, H.F. Ates, I. Hamzaoglu, "A high performance hardware architecture for an SAD reuse based hierarchical motion estimation algorithm for H.264 video coding," IEEE Int. Conf. Field Programmable Logic and Applications, Aug. 2005, pp. 509-514.
- [6] JVT Reference Software Coordination, appeared in <http://iphome.hhi.de/suehring/tml/>
- [7] M. Melani, L. Fanned, S. Saponara, "Algorithmic/architectural design for H.264/MPEG-4 AVC low-power video codec," PhD Research in Microelectronics and Electronics, Vol. 1, Jul. 2005, pp. 209-212.
- [8] Gwo Giun Lee, K.A. Vissers, Bin-Da Liu, "On a 3D recursive motion estimation algorithm and architecture for digital video SoC," IEEE Int. Conf. 47th Midwest Symp. Circuits and Systems, MWSCAS '04, Vol. 2, Jul. 2004, pp. II/449-II/451.
- [9] T.Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. Sullivan, "Rate-constrained coder control and comparison of video coding standards," IEEE Trans. Circuit Syst. Video Technol., vol. 13, no. 7, pp. 688-703, July 2003.
- [10] H.264/MPEG-4 AVC, in <http://en.wikipedia.org/wiki/H.264>
- [11] C. Gomila, P. Yin, "New features and applications of the H.264 video coding standard," Proc. IEEE Int. Conf. Information Technology: Research and Education, ITRE2003, Aug. 2003, pp. 6-10.
- [12] Quartus II Handbook Volume 5: Embedded Peripherals, in <http://www.altera.com/literature/lit-nio2.jsp>