

Survey on H.264 Standard

R. Vani¹ and M. Sangeetha²

¹ Anna University of Technology, Chennai

² ECE Dept., Karpaga Vinayaga College of Engineering and Technology, Kanchipuram
{vani_gowtham, sang_gok}@yahoo.com

Abstract. The progress of science and technology demands multimedia applications to be realized on embedded systems as it involves transfer of large amounts of data. Compared with standards such as MPEG-2, MPEG-4 Visual, H.264 can deliver better image quality at the same compressed bit rate or at a lower bit rate. The increase in compression efficiency and flexibility come at the expense of increase in complexity, which is a fact that must be overcome. Therefore, an efficient Co-design methodology is required, where the encoder software application is highly optimized and structured in a very modular and efficient manner, so as to allow its most complex and time consuming operations to be offloaded to dedicated hardware accelerators. This paper provides an overview of the features of H.264 and surveys the emerging studies related to new coding features of the standard.

Keywords: H.264, Motion Estimation, Co-design, Hardware accelerators, Optimization.

1 Introduction

The H.264 Advanced Video Codec is an ITU standard for encoding and decoding video with a target coding efficiency twice that of H.263 and with comparable quality to H.262. MPEG-4 was launched to address a new generation of multimedia applications and services such as interactive TV, internet video etc. An increasing number of services and growing popularity of HDTV are creating much more need for higher coding efficiency. Another name for H.264 is MPEG-4 Advanced Video Coding (AVC) standard. Since the standard is the result of collaborative effort of the VCEG and MPEG standards Committees, it is informally referred to as Joint Video Team (JVT) standard as well [8]. Applications such as internet multimedia, wireless video, personal video recorders, video-on-demand and videoconferencing have an inexhaustible demand for much higher compression to enable best video quality as possible [27]. Ongoing applications range from High Definition Digital Video Disc (HD-DVD) or BluRay for living room entertainment with large screens to Digital Video Broadcasting for Handheld terminals (DVB-H) with small screens [13]. The H.264 standard is a new state of video coding standard that addresses aforementioned applications with higher compression than earlier standards.

It enables PAL 720×576 resolution video to be transmitted at 1Mbit/sec. According to the instruction profiling with HDTV1024P (2048×1024 , 30fps) specification, H.264/AVC decoding process requires 83 Giga-Instructions Per Second (GIPS) computation and 70 Giga-Bytes Per Second (GBPS) memory access. As for H.264/AVC encoder, up to 3600 GIPS and 5570 GBPS are required for HDTV 720P (1280×720 , 30fps) specification. The increasing video resolutions and the increasing demand for real-time encoding require the use of faster processors. However, power consumption should be kept to a minimum. Therefore, for real-time applications, accelerating by the dedicated hardware is a must. This paper provides an overview and summarizes emerging studies on the coding features of the H.264 standard.

The paper is organized as follows: Section 2 presents an overview of the H.264 standard. It provides details of coding structure of H.264. Following sections highlight some key technical features that enable improved operation of H.264 for broad variety of applications. Section 3 examines new algorithms for variable block size matching algorithm for the Motion Estimation. Section 4 provides information on different scanning methods and search patterns. Section 5 emphasizes co-design and co-simulation approaches. Section 6 elaborates the need for optimization and advance features. Finally, in Section 7 concluding remarks are made.

2 Overview of the H.264 Standard

Video compression efficiency achieved in H.264 standard is not a result of single feature but rather a combination of a number of encoding tools. Figure 1 depicts the structure of H.264/AVC video encoder [24]. The H.264/AVC encoder contains three steps: prediction, transformation/quantization and entropy encoding. In H.264/AVC, Macro block mode decision and Motion Estimation are the most computationally expensive processes. Mode decision is a process such that for each block-size, bit-rate and distortion are calculated by actually encoding and decoding the video. Therefore, the encoder can achieve the best Rate Distortion (RD) performance, at the expense of calculation complexity [27].

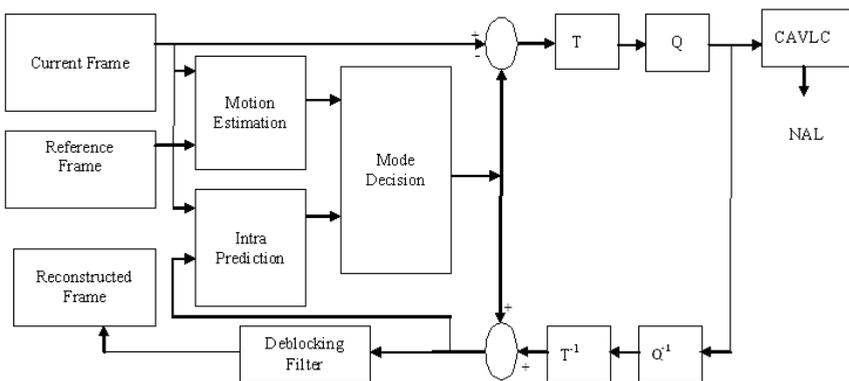


Fig. 1. H.264 Encoder Block Diagram (T-Transform Q-Quantization)

Motion Estimation

An important coding tool of H.264 is the variable block size matching algorithm for the ME (Motion Estimation) [3,18] which is part of the prediction step. In H.264 encoder, the frame is divided in 16×16 pixel macroblocks. The motion estimator has two inputs: a macroblock (MB) from the current frame and a 48×48 pixel search area (SA) from the previous frame. For each MB in the current frame, a search window is defined around a point in the reference frame. A distortion measure is defined to measure the similarity between the candidate MB and the current MB. A search is performed within the search window for the best matched candidate MB with maximum similarity. The displacement of the best matched MB from the current MB is the Motion Vector (MV).

Transformation/Quantization

In [26], H.264 uses three transforms depending on the type of residual data that is to be coded: Hadamard transform for the 4×4 array of luma DC coefficients in Intra- 16×16 mode, a Hadamard transform for the 2×2 array of chroma DC coefficients and a DCT-based integer transform for all other 4×4 blocks in the residual data. By using Integer transformation, inverse-transform mismatches are avoided. A quantization parameter (QP) is used in quantization process which can take 52 different values on a macroblock basis. These values are arranged so that an increase of one in QP means an increase of quantization step size by approximately 12%. Rather than constant increment, the step sizes increase at a compounding rate. This feature is not present in prior standards and it is of great importance for compression efficiency.

Entropy Encoding

In H.264, two methods of entropy coding are supported. The first one is Context-Adaptive Variable Length Coding (CAVLC) and the other one is Context-Adaptive Binary Arithmetic Coding (CABAC). In CAVLC, entropy coding performance is superior to the schemes using a single VLC table. CABAC improves the coding efficiency further (approximately 5–15% bit saving) by means of context modeling which is a process that adapts the probability model of arithmetic coding to the changing statistics within a video frame. It is observed from [28] that the encoding time is significantly shorter for CABAC (21s less with a bit-rate of 300 kbps and 43s less with a bit-rate of 1 Mbps) whereas the decoding time is reduced slightly with CAVLC (1.2s less with a bit-rate of 300 kbps and 1.5s less with a bit-rate of 1 Mbps). Concerning the visual quality, the average Y-PSNR is better with CABAC (1.3% better with a bit-rate of 300 kbps and 1.1% better with a bit-rate of 1Mbps).

Inverse Transformation and Quantization

Since residual data exhibits high spatial entropy, H.264 employs a lossy low-pass discrete cosine transform to develop a compact representation of the residual values.

H.264 also allows variable quantization of DCT coefficients to enhance coding density. In [29], the mapping of a two-dimensional inverse discrete cosine transform (2-D IDCT) onto a word-level reconfigurable Montium processor is described. It shows that the IDCT is mapped onto the Montium tile processor (TP) with reasonable effort and presents performance numbers in terms of energy consumption, speed and silicon costs.

Intraprediction

Video frames have a high amount of spatial similarity. Intraprediction use previously decoded, spatially-local macroblocks to predict the next macroblock and it works well for low-detail images [18].

Interprediction

Video frames nearby in time have only small differences. It attempts to capitalize on this similarity by encoding macroblocks in the current frame using a reference to a macroblock in a previous frame and a vector representing the movement that macroblock took to a 1/4 pixel granularity. The decode uses an interpolation process known as motion compensation to generate the prediction value. Fractional motion vectors are interpolated from multiple previous macroblocks.

Deblocking Filter

Lossy compression is used to encode pixel blocks in H.264 and decoding errors appear most visibly at the block boundaries. To remove these visual artifacts, the H.264 CODEC incorporates a smoothing filter into its encoding loop. H.264 also incorporates fine-grained filter control to preserve these edges. With the filter, the blockiness is reduced, while the sharpness of the content is basically unchanged and the subjective quality is significantly improved. The filter reduces bit rate typically by 5-10% compared to the non-filtered video.

3 Motion Estimation Algorithms

Variable Block Size (VBS) ME allows different MVs for different sub-blocks and can achieve better matching for all sub-blocks and higher coding efficiency than Fixed Block Size ME (FFBSME). It is especially useful for MBs containing multiple objects each with possibly different motion and it can also be useful for MBs with rotation and deformation. VBSME has good RD performance compared with FBSME, but it has huge computational requirement and irregular memory access making it hard for efficient hardware implementation. The H.264 in [19] allows a 16×16 MB to be partitioned into seven kinds of sub-blocks as shown in Figure 2.

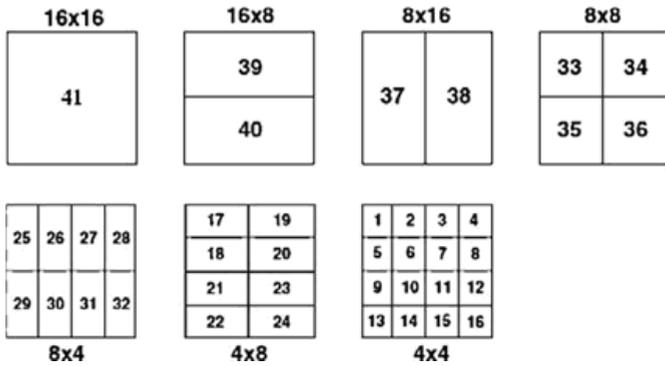


Fig. 2. Variable block size in H.264/AVC

In [9], three key points are observed for deriving an efficient ME algorithm from optimization theory.

1. The initial search point should be as close to optimal solution as possible. This goal can be achieved by exploiting the spatio-temporal correlation of MV fields.
2. An efficient update process is necessary to limit the number of SPs or iterations within an acceptable extent.
3. Multiple update paths induced by multiple initial points prevent local minimal trapping on multimodal error surface.

In an Adaptive Crossed Quarter Polar Pattern Search algorithm [6], an H.264 compatible median vector predictor (MVP) is generated for determining the initial search range. The direction of the pattern is adaptively selected with a shape of the quarter circle. The length (radius) of the search arm is adjusted to improve the search. Procedure of algorithm involves four steps:

1. Get a predicted MV (MVP) for the current block.
2. Find the direction of a search pattern, determine the pattern size "R", choose initial search point (SPs) along the quarter circle and extended predicted MV, together with the point of current block (0, 0) and MVP
3. Check the initial SPs, and get an minimum matching error point (MME) which has the minimum sum of absolute differences (SAD).
4. Refine the initial search by applying the unit-sized square pattern to that MME point and successive MME points iteratively, and find a final MV for the current block, corresponding to the final best matching point is identified.

In Ultra Low-Complexity Fast VBSME fast VBSME algorithm in [31] is described as follows, which adopts the CDS search strategy, and the SAD is replace by the pixel decimated SAD (PDSAD).

1. Cross-search: one cross search pattern with 9 search points is adopted. If the found minimum MV occurs in the cross center, this algorithm stops.
2. Half diamond search: two extra search points which are the nearest to the current minimum are checked. If the found minimum is still located in the middle of the cross pattern, namely $(\pm 1, 0)$ or $(0, \pm 1)$, the algorithm stop.

3. Large diamond search: the current minimum search point is used as the search center and the large diamond pattern with 9 search points is used to trace the motions. This step continues until the found minimum is located in the diamond center.
4. Small diamond search: the minimum search point in previous step is used as the search center, and one small diamond pattern with 4 search points is then adopted to refine the search result. The final minimum is returned as the best MV.

In [30], Multi-pass and frame parallel algorithms are proposed to accelerate various motion estimation (ME) tools in H.264 with the graphics processing unit (GPU), GeForce 7800 GT. Compared to implementations with CPU, about 6 times to 56 times speed-up can be achieved for different ME algorithms. GPU is parallel architecture and it is able to efficiently process motion estimation. First an algorithm is proposed to map motion estimation (ME) on generic GPU to accelerate video encoding. Second, advanced motion estimation algorithms in H.264, such as quarter-pel ME and multiple reference frame ME, are implemented. ME contains mainly two parts: integer ME (IME) and fractional ME. Runtime profiling of H.264 JM encoder reveals that IME consumes close to 60% of total encoder time and up to 90% when fractional ME is included. Thus efficient ME algorithms and hardware architectures for IME are needed. IME architecture in [2] shows high throughput and it's a cost efficient VLSI architecture for integer full-search VBS-ME. In an efficient FME implementation, the trade-off among processing time, memory access data bus and hardware utilization should be balanced. According to [22], IME and FME must be computed in 1025 cycles which will affect the efficiency of the hardware implementation. IME is performed prior to FME in which integer pixel search tries to find the best matching integer position and the best integer pixel motion vectors (MV) are determined by using a performance cost metric. Then, FME performs a half-pixel refinement about the integer search positions and then a quarter-pixel one is performed around the best half-pixel positions. As a result, pipeline architecture is a must to implement IME and FME. The p264 platform [16] is a configurable software application derived from version JM14.0 of the H.264/AVC Reference Software Model that presents a highly modular and flexible structure where all the functional modules of the video encoder are implemented as independent and self-contained software modules. It allows replacing a software realization of any given function of the video encoder by a system call to a hardware accelerator implementing that same function whenever higher performance levels are required. Some algorithms perform best on fine-grain reconfigurable architectures whereas others perform better on coarse-grain reconfigurable or general purpose processing (GPP) tiles [25].

4 Scanning Methods

Different scanning methods and search patterns are discussed in this section. In [3,4,12,31] different Search patterns full search (FS), 3 step search (3SS), 4-step search (4SS), diamond search (DS), cross-diamond search (CDS), and hexagon search (HEXBS) are discussed. In [4,31] 3SS yields better speedup when compared to FS,

DS ME algorithms, by taking a Leon3 uniprocessor video encoding system as the reference platform. The quality of fast ME algorithms have the following relations: $DS \approx CDS > 4SS > HEXBS$. It is observed that the diamond pattern in DS and CDS is more accurate than the rectangle pattern in 4SS and the hexagon pattern in HEXBS. It is desirable to employ different search patterns, i.e, adaptive search patterns, for a variety of the estimated motion behaviors. An adaptive search patterns [6] is devised to detect the optimal or sub-optimal search points in the initial stage. The idea is to choose some initial search points (SPs) along the pattern to be checked in the initial search range. To reduce the number of initial SPs and keep the good probability of obtaining best matching point which has the minimum SAD, a fractional (quarter) polar search pattern is designed. The direction of search pattern is defined by the direction of a quarter circles which comes from the predicted motion vector (MV). Figure 3 displays the possible patterns adaptively that employ the directional information of a predicted MV to increase the possibility of acquiring the optimal minimum matching error (MME) point for refined search. The radius of a designed pattern, is defined as

$$R = \text{Max}\{|\text{PredMV}_y|, |\text{PredMV}_x|\}$$

where R is the radius of quarter circle and PredMV_y, PredMV_x the vertical and horizontal components of the predicted MV respectively.

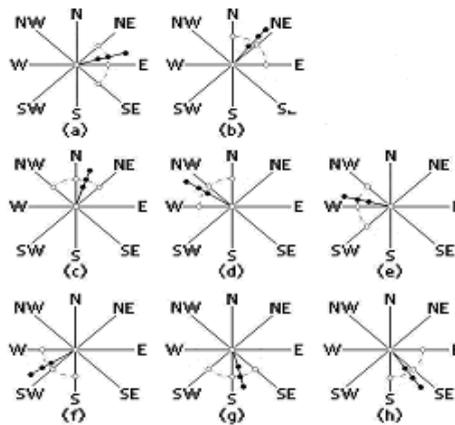


Fig. 3. Possible adaptive search patterns

In Raster Scan, the search locations in the first row are scanned from left to right, followed by the second row from left to right, and so on. Raster Scan is effective in reusing data horizontally with relatively high data re-use ratio but with redundant loading. The data re-usability is improved slightly in some architecture by another scanning order called Snake Scan as shown in Fig. 4(a). Snake Scan processes the first row from left to right, then the second row from right to left, and then the third row from left to right, and so on. In both Raster Scan and Snake Scan, the data re-use

ratio and search window size is fixed. A novel scanning order called Smart Snake (SS) is proposed in [1] which can achieve variable data re-use ratios and minimum redundant data loading. Search window is divided into an array of non-overlapping rectangular sub-regions that span the search window which is shown in Fig. 4(b). In each rectangular sub-region, Snake Scan is performed to achieve significantly higher data re-use. After one sub-region is searched, it will move into an adjacent region and Snake Scan will be applied again. In different sub-regions, Snake Scan may be performed from top to bottom (L1), or from bottom to top (L2). It may start from left and end at right (L1, L2, L3), or start from right and end at left (L4, L5, L6). It may be horizontal (L1, L2) or vertical (L3, L4). "Horizontal" to mean the original Snake Scan which processes the search points row-by-row and "vertical" to mean column-by-column Snake Scan. The width (or the height) of each sub-region is restricted to be less than or equal to a parameter M.

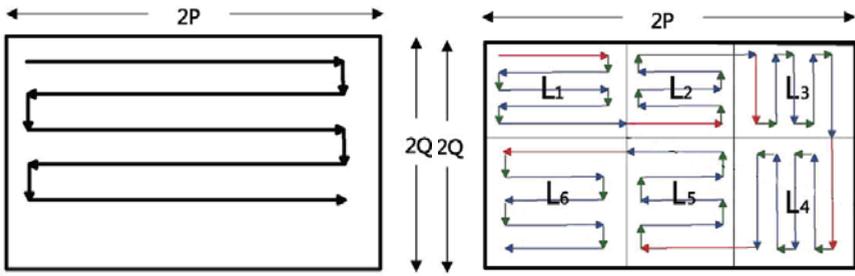


Fig. 4. (a) Snake Scan, (b) Smart Snake Scanning order (SS)

A new scan order for reference datas writing and reading is introduced in [2] to improve the efficiency of memory accessing and to obtain high data-reuse of the search area. The architecture of VBS-ME allows the real-time processing of 1280×720 at 38 fps with FS-BMA in a search range $[-32, +32]$ with 36k gate counts. Processing pipelining of 4×4 SAD Parallel processing and pipelining techniques are used to reduce the latency and increase the data utilization

5 Co-design and Co-simulation Approaches

The emphasizes of Co-Design is on the area of system specification, hardware software partitioning, architectural design, and the iteration in between the software and hardware as the design proceeds to next stage. The hardware and software co design makes it possible. A Multi Core H.264 video encoder is proposed in [4,5,23], by applying a novel hardware software co-design methodology which is suitable for implementing complexity video coding embedded systems. The hardware and the software components of the system are designed together to obtain the intended performance levels. At the hardware level, the designer must select the system CPU, hardware accelerators, peripheral devices, memory and the corresponding interconnection structure. The software component addresses the design of a program

to efficiently implement the application algorithms and to support the communications between all the system hardware components. The code is further optimized by taking into consideration the characteristics of the hardware components and by applying the most complex and efficient modes of the software compiler tools. In H.264 video decoder, different blocks can be partitioned into several stages. The implementation of each function under different partitions is shown in Table 1 and it is observed that architecture for partition 4 can achieve more than three times acceleration in performance.

Table 1. Implementation of each function under different partitions

	VLD	IQ	IT	Intra	Inter	Reconst.	DB
Partition 1	SW	SW	SW	HW	HW	SW	SW
Partition 2	SW	SW	SW	HW	HW	HW	SW
Partition 3	HW	HW	HW	HW	HW	HW	SW
Partition 4	SW	HW	HW	HW	HW	HW	HW

Two co-design approaches were identified in [25]. The first co-design approach is shown in figure 5a, to develop both the software and hardware separately. Verification does not take place until the design is deployed to a specific hardware platform which leads to late detection of mistakes in the HW/SW partitioning and implementation. In the second approach, all subsystems are verified in one environment and it becomes a difficult task. One method is to represent all systems in one HDL, which can involve model degradation. The second method is to use a simulator that supports all different HDLs used and the third method is to use different simulators for each system and verify the integrated system using co-simulation. Co-simulation is useful in HW/SW co-design [20,21]. The co-simulation design approach is depicted in figure 5b. Co-simulation can be done by either connecting two simulators known as direct coupling or by the use of a co-simulation backplane. The co-simulation allows for designing in much short iteration while verifying functional behaviour.

Co-design is proposed in [14] as a chip named OR264 with mixed flexibility and it is partitioned that the hardware is used to boost the performance and efficiency of key operations. The chip is fabricated using hardware software architecture to combine performance and 0.1 8-ptm 6-layers metal CMOS process in UMC. It contains 1.5M transistors and 176k bits embedded SRAM and can operate at 100MHz. The die size of the processor is 4.8mm × 4.8mm and the critical path delay is 10ns. Results evidence the low hardware requirements and prove that real-time computation of MVs for QCIF video sequences with only one ME IP core is possible. Data Exchange Mechanism (DEM) controller is the only one master in the architecture used in the co design for H.264 Video Decoder [7] and the other hardware accelerators are all slaves. DEM controller dominates all the I/O access of the hardware accelerators and on the other hand, it will also dispatch the data and the parameters passed by the processor to the corresponding hardware accelerators. As a result, users can add or delete hardware accelerators easily since there is no data dependency among hardware accelerators.

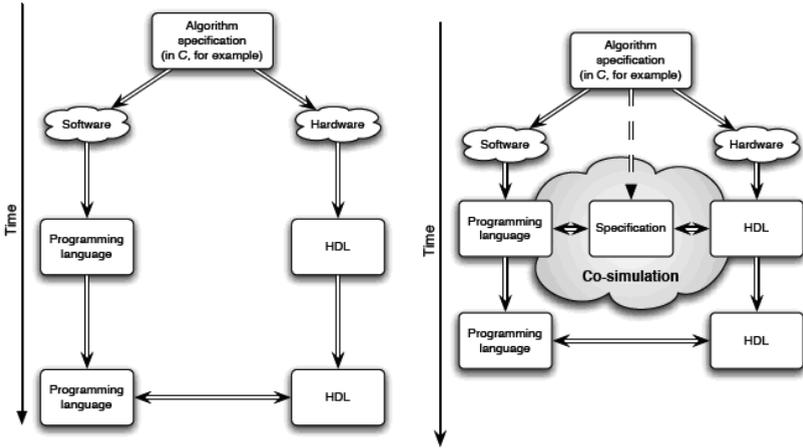


Fig. 5. Co-design approaches (a) Traditional Approach (b) Co-simulation approach

6 Optimization and Other Features

Mismatch measures such as sum of absolute difference (SAD), sum of squared difference (SSD) and sum of absolute transformed difference (SATD) are available and in which SAD is most common due to its simplicity and effectiveness [1]. Most existing hardware ME architectures are based on SAD. In the JM version 15.1 of the H.264 reference software, the ME chooses the best mode by using a Lagrangian mode decision to compute an estimation of the bits required to code MVs. For each subblock of a MB, Langrangian cost (J) defined as

$$J = SAD + \lambda MVcost(MV_{cur} - MV_{pred})$$

where $MVcost$ represents the number of bits required to code the difference of current MV (MV_{cur}) and motion prediction (MV_{pred}) and λ is the Langrangian multiplier. An alternative measure is called rate-distortion (RD) cost function which is given by

$$RDCost = D + \lambda \cdot R$$

where D is the distortion such as SSD, SATD, or SAD, R is the associated bit rate and λ is the Lagrangian multiplier. Recent ME algorithms tend to use RD cost due to its superior performance. RDOMFS circuit with a small search range can achieve better RD performance with low power consumption than FS-SAD. It is hard to design efficient hardware for RD for at least two reasons. First, RD computation requires floating point operation for the multiplication of and R which is time and resource consuming. If this is to be relieved by using lookup tables, it would require huge chip area for the lookup tables. Second, the data flow in the computation of MV_{median} is irregular and requires a large amount of on-chip memory to store the required past MVs. As a result of microarchitectural change, the deblocking filter implementation

in [8] decreases area dramatically from 2.74 mm² to 0.69 mm². Optimized deblocking filter yields 12% increase in throughput of the entire design, and thereby reducing the design critical path by 35%. The 4:1 Haar filter based pixel decimation is adopted to reduce matching costs. In FFSBM proposed in [19], filter reduces bit rate typically by 5–10% compared to the non-filtered video.

In [3,15], it was observed that results obtained with the implementation of a multicore SoC of an H.264/AVC video encoder in a Virtex4 FPGA demonstrated that speedups greater than 15 can be obtained for the ME task and over 3 for the global encoding operation. Huge reduction in the computation time of the ME operation and transfer times for the pixel data (MB and SA) and for the ME results (MVs) are negligible and it's about 0% of the total encoding time. An efficient quarter pel ME hardware [17] is designed for portable applications together with half pel ME. This architecture is implemented in VHDL and it was found that code works at 60MHz in a Xilinx Virtex II FPGA. The performance results [25] shows that FPGA implementation shows a speed-up of 43.6 whereas the Montium implementation shows a speedup of 21.5, both compared to a software-only implementation. The speedup values validate the adopted methodology and hardware software design partitioning.

Adaptivity in search patterns [12], will greatly reduce the dynamic complexities of motion estimation and real time encoding of 1280 × 720 video can be processed at 30 fps. Reconfigurable architecture for Standards MPEG2, MPEG4, H.263 and H.264 [10] requires more power and silicon area to achieve flexibility. Configurable architecture adopting a DEM [7] controller to fit the best tradeoff between performance and cost when realizing H.264 video decoder for different applications. The FPGA implementation can process 34 VGA frames (640 × 480) per second. It reduces the amount of computation and thereby reducing power consumption. In [24], Montium target platform consists of an ARM946E-S and a Xilinx Virtex XC2V8000 FPGA containing the Montium TP. The clock frequency is 100 MHz, both for the ARM and the Montium. The number of clock cycles needed to process a macroblock is always the same. There are two important observations underlying the main idea of the algorithm of [31] in which the first is direct pixel decimation is not suitable for H.264/AVC because of the small Sub-block sizes. Second is by adopting low-pass filter based pixel decimation, the original SAD operation can be reduced to 25%, reduce the computation to about 0.2% of FFS, average PSNR loss and bit rate increase are 0.12dB and 2.81%, respectively and still maintaining robust image quality. The ACQPPS architecture [6] can yield better performance in terms of average PSNR of -0.05dB, +0.34dB and +0.11dB.

It is observed that under the CBR mode, encoding time using CABAC is superior over CAVLC, whereas the degradation in decoding time is insignificant [28]. The most striking results in the context of VBR, Rate Distortion Optimization (RDO) provides better visual quality but the encoded video is bigger in size and the time needed for encoding is longer. RDO is well suited for broadcasting of prerecorded high quality videos but should be avoided for low delay applications. Testing RDO under CBR is a good means to determine if the improvements in terms of visual quality remain when the sizes of the encoded videos are the same. A second

experiment has been carried out to test the performances of RDO under CBR. CBR and RDO can be considered as complementary tools. If the bit-rate is fixed and the QP values are determined by the rate-control algorithm, RDO simply determines the best prediction mode. Particle swarm optimization algorithm (PSO) is presented for multi fusion images in which size of the blocktype regions are optimized. Experiments are conducted on both artificial and natural multi focus images and results show that PSO based method outperforms Laplacian pyramid transform, Discrete Wavelet Transform and Genetic algorithm in terms of quantitative and visual evaluations [26].

VLSI architecture for FME with processing capacity for 1080HD real-time video streams with three different pipelined processors a high throughput and low area cost , which can generate the residual image and the best MVs to be encoded. In [12], better area/throughput is achieved by exactly choosing the Processing Element Array size to reduce the Gate Count and the bandwidth lower bound and upper bound for FS, 3SS, and HS is calculated. To give an accurate comparison, the Gate Count vs Throughput ratio (GCTR) is defined as

$$\text{GCTR} = \text{Gate Count} / \text{Throughput} = (\text{Gate Count}) * \text{Cycle per MB}$$

and it is found that lower GCTR indicates higher hardware efficiency. With the advent of 3G, Optimization of H.264 codec and improvement of mobile system will have significant improvement [11].

7 Conclusion

H.264/AVC represents a major step in the development of video coding standards in terms of both coding efficiency enhancement and flexibility for effective use over a broad variety of network types and application. Co-Design approaches can be used to explore Motion estimation algorithms to yield better timing and speed optimization by using particle swarm optimization, simulated annealing and other methods. Co-Simulation tools to be used for further enhancing speed and reducing power consumption. Fine grain partitioning may be done for every modules of video codec to reduce the area. Development of the encoder conforming to the standard is still considered to be a challenging issue, particularly for real-time applications. The future design methodologies and associated tools must provide both modular refinement and high-level synthesis.

References

- [1] Wen, X., Fang, L., Li, J.: Novel RD-Optimized VBSME with Matching Highly Data Re-Usable Hardware Architecture. *IEEE Transactions On Circuits And Systems For Video Technology* 21(2), 206–219 (2011)
- [2] Gu, M., Yu, N., Zhu, L., Wenhua: High Throughput and Cost Efficient VLSI Architecture of Integer Motion Estimation for H.264/AVC. *Journal of Computational Information Systems*, 1310–1318 (2011)

- [3] Dias, T., Roma, N., Sousa, L.: H.264/AVC framework for multi-core embedded video encoders. In: International Symposium on System on Chip (SoC), pp. 89–92 (2010)
- [4] Dias, T., Roma, N., Sousa, L.: Hardware/Software Co-Design Of H.264/AVC Encoders For Multi-Core Embedded Systems. In: Conference on Design and Architectures for Signal and Image Processing (DASIP), pp. 242–249 (2010)
- [5] Dias, T., Roma, N., Sousa, L.: Hardware/Software Co-Design Of H.264/AVC Encoders For Multi-Core Embedded Systems: inescid, ISEL (2009)
- [6] Qiu, Y., Badawy, W.: The Hardware Architecture Of A Novel Motion Estimator With Adaptive Crossed Quarter Polar Search Patterns For H.264. In: Canadian Conference on Electrical and Computer Engineering, CCECE 2009, pp. 819–822 (2009)
- [7] Jian, G.-A., Chu, J.-C., Huang, T.-Y., Chang, T.-C., Guo, J.-I.: A System Architecture Exploration on the Configurable HW/SW Co-design for H.264 Video Decoder. In: IEEE International Symposium on Circuits and Systems, ISCAS 2009, pp. 2237–2240 (2009)
- [8] Fleming, K., Lin, C.-C., Dave, N., Arvind, Raghavan, G., Hicks, J.: H.264 Decoder: A Case Study in Multiple Design Points. In: 6th ACM/IEEE International Conference on Formal Methods and Models for Co-Design, MEMOCODE 2008, pp. 165–174 (2008)
- [9] Lee, G.G., Wang, M.-J., Lin, H.-Y., Su, D.W.-C., Lin, B.-Y.: Algorithm/Architecture Co-Design of 3-D Spatio-Temporal Motion Estimation for Video Coding. *IEEE Transactions on Multimedia* 9(3), 455–465 (2007)
- [10] Lu, L., McCanny, J.V., Sezer, S.: Reconfigurable ME Architecture for Multi-standard video compression. In: IEEE International Conf. on Application-specific Systems, Architectures and Processors, 2007 ASAP, pp. 253–259 (2007)
- [11] Wang, S.-F., Huang, Z.-Q., Hou, Y.-B.: A Design of Low-cost, Low-bandwidth Mobile Video Surveillance System Based on DM6446. In: International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007, pp. 3079–3083 (2007)
- [12] Zhang, L., Gao, W.: Reusable Architecture and Complexity-Controllable Algorithm for the Integer/Fractional Motion Estimation of H.264. *IEEE Transactions on Consumer Electronics*, 749–756 (2007)
- [13] Chen, T.-C., Lian, C.-J., Chen, L.-G.: Hardware Architecture Design of an H.264/AVC Video Codec. In: Conference on Asia and South Pacific Design Automation, p. 8 (2006)
- [14] Yang, K., Zhang, C., Du, G., Xie, J., Wang, Z.: A Hardware-Software Co-design for H.264/AVG Decoder. In: IEEE Asian Solid-State Circuits Conference, pp. 119–122 (2006)
- [15] Le, T.M., Tian, X.H., Ho, B.L., Nankoo, J., Lian, Y.: System-on-Chip Design Methodology for a Statistical Coder. In: Seventeenth IEEE International Workshop on Rapid System Prototyping, pp. 82–90 (2006)
- [16] Rodrigues, A., Roma, N., Sousa, L.: p264: Open platform for designing parallel H.264/AVC video encoders on multi-core systems. In: International Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 81–86 (2010)
- [17] Lin, C.-C., Lin, Y.-K., Chang, T.-S.: A Fast Algorithm and Its Architecture for Motion Estimation in MPEG-4 AVC/H.264 Video Coding. In: IEEE Asia Pacific Conference on Circuits and Systems, pp. 1248–1251 (2006)
- [18] Sayood, K.: Introduction to Data Compression, 3rd edn. Elsevier Publishers (2006)

- [19] Zhang, L., Gao, W.: Improved FFSSBM Algorithm and its VLSI Architecture For Variable Block Size Motion Estimation Of H.264. In: Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems, pp. 445–448 (2005)
- [20] Hardware/Software Co-design of the H.264/AVC standard. In: Fifth FTW PhD Symposium, Faculty of Engineering, Ghent University, Paper no.120 (2004)
- [21] De Vleeschouwer, C., Nilson, T., Denolf, K., Bormans, J.: Algorithmic and Architectural co-design of a motion-estimation engine for low power devices. *IEEE Transactions on Circuits and Systems for Video Technology*, 1093–1105 (2002)
- [22] Ruiz, G.A., Michell, J.A.: An Efficient VLSI Architecture of Fractional Motion Estimation in H.264 for HDTV. *J. Sign. Process. Syst.* (2010)
- [23] Kalavade, A., Subramanyam, P.A.: Hardware/ Software partitioning for multi function systems. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 17(9), 819–837 (1998)
- [24] Oktem, S., Hamzaoglu, I.: An efficient Hardware architecture for Quarter Pixel Accurate H.264 Motion Estimation. In: 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, pp. 444–447 (2007)
- [25] Colenbrander, R.R., Damstra, A.S., Korevaar, C.W., Verhaar, C.A., Molderink, A.: Co-design and Implementation of the H.264/AVC Motion Estimation Algorithm using co-simulation. In: 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools, pp. 210–215 (2008)
- [26] Aslantas, V., Kurban, R.: Extending depth of field of a digital camera using particle swarm optimization based image fusion. In: *IEEE 14th International Symposium on Consumer Electronics (ISCE)*, pp. 1–5 (2010)
- [27] Ozbek, N., Tunali, T.: A Survey on the H.264/AVC Standard. *Turk. J. Elec. Engin.* 13(3) (2005)
- [28] Mazataud, C., Bing, B.: A Practical Survey of H.264 Capabilities. In: Seventh Annual Communication Networks and Services Research Conference, pp. 25–32 (2009)
- [29] Smit, L., Rauwerda, G., Molderink, A., Wolkotte, P., Smit, G.: Implementation of a 2-D 8×8 IDCT on the reconfigurable Montium core. In: *International Conference on Field Programmable Logic and Applications*, pp. 562–566 (2007)
- [30] Lee, C.-Y., Lin, Y.-C., Wu, C.-L., Chang, C.-H., Tsao, Y.-M., Chien, S.-Y.: Multi-Pass and Frame Parallel algorithms of Motion Estimation in H.264/AVC for generic GPU. In: *International Conference on Multimedia and Expo.*, pp. 1603–1606 (2007)
- [31] Song, Y., Liu, Z., Ikenaga, T., Goto, S.: Ultra low complexity fast Variable Block Size Motion Estimation algorithm in H.264/AVC. In: *IEEE International Conference on Multimedia and Expo.*, pp. 376–379 (2007)