A LOW-COMPLEXITY HARDWARE-ORIENTED MODE DECISION SCHEME BASED ON RATE-DISTORATION ESTIMATION

Meng Li, Chuang Zhu, Yuan Li, Xiaofeng Huang, Huizhu Jia, Xiaodong Xie, Wen Gao, Fellow IEEE
National Engineering Laboratory for Video Technology
Peking University, Beijing, China
{limeng, czhu, yuanli, xfhuang, hzjia, xdxie, wgao}@jdl.ac.cn

ABSTRACT

Video compression plays an important role in mobile applications, because more and more people use video to communicate with each other (like video call etc). However, the resources (energy, memory etc.) on mobile devices are limited, thus how to achieve a high coding performance in these devices becomes a big challenge. The recent standards such as H.264, HEVC and audio video coding standard (AVS) employ Rate distortion optimization (RDO) to select the best coding modes, however it results in extremely high computational complexity. This work presents a hardware friendly mode decision (MD) scheme. First, hardware-oriented *RDcost* estimation method is proposed by using least squares technique to reduce computational burden of RDO-based MD. Second, reconstructed-original (REC-ORG) united intra prediction scheme is presented to break the data dependency, while maintaining high coding performance. Third, highly efficient MD pipeline architecture is put forward to enhance MD processing capacity. The coding efficiency of our adopted MD scheme far outperforms (0.402 dB PSNR gain in average) the traditional SAD methods and the throughput of our designed pipeline is increased by 29%, 23% and 23% for I, P and B frames, respectively, compared with the existed RDO-based architecture.

***Index Terms*—**mobile device, mode decision, hardware friendly, RDcost estimation, pipeline

1. INTRODUCTION

With the popularity of smartphones and laptops, mobile applications are increasing sharply these years. Video compression plays an important role in these mobile applications, because more and more people use video to communicate with each other. However, the resources (energy, memory etc.) on mobile devices are limited, thus how to achieve a high coding performance in these devices becomes a big challenge. RDO technique greatly improves coding efficiency [1]. RDO-based mode decision (MD) is adopted in many video systems with the increasing demands for high quality video applications. But the mode decision unit has to fully perform discrete cosine transform (DCT), quantization (Q), inverse quantization (IQ), zigzag scanning (ZIGZAG), entropy coding (EC), inverse discrete cosine (IDCT) transform, and pixel reconstruction (REC) to obtain the accurate entropy coding bits (R) and reconstruction distortion (D) for each mode in MD process. Besides, more and more modes are adopted in recent video coding standards, such as AVS [2], in which there are 5 prediction modes for intra luma block (Vertical (V), Horizontal (H), DC, Diagonal down right (DR) and Diagonal down left (DL)) and 4 modes for each chroma block (Vertical, Horizontal, DC, and Plane (P)). AVS standard also uses variable block size (VBS) ME and MC techniques, which results in different inter modes (16×16, 16×8, 8×16, 8×8) to be chosen. In addition, direct and skip modes are adopted in AVS. H. 264 [3] and HEVC [4] standards adopt more intra modes and more VBS partitions to optimize coding performance of the encoder. The abundant modes and the complexity computation process for each mode makes the total RDO-based MD computation process unbearable in many real time video coding applications, as shown in Figure 1 (taking AVS for example).

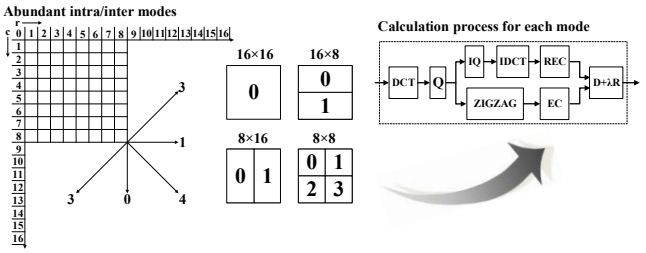

Fig. 1. Abundant intra/inter modes and R-D calculation process for each mode in RDO-based MD.

A number of strategies have been developed in recent years to alleviate the unbearable computational burden in RDO-based MD. Some papers [5]-[6] have presented some fast MD methods trying to reduce the candidate modes to achieve the goal. These algorithms firstly study the spatial features or temporal features of a block and then skip the unnecessary candidate modes based on experimental results. Other papers [7]-[8] did not skip any candidate modes and focused on reducing the calculating complexity of each R-D cost value. In [8], Xin et al. deduced a novel weighted sum of quantized transform coefficients and utilized it as an efficient rate estimator; the authors also proposed a new transform-domain distortion (TDD) estimation method using the discarded lower bits in the quantization process.

This work is partially supported by grants from the Chinese National Natural Science Foundation under contract No.61171139 and No. 61035001, and National High Technology Research and Development Program of China (863 Program) under contract No.2012AA011703.

Hardware encoding solution may be a better choice in resource limited applications (smartphones, iPads, etc.). However, most of the works mentioned above are not hardware friendly and generally implemented on software platform. Among them, paper [8] achieves outstanding coding performance, but the rate-distortion estimation calculation process is not hardware friendly. Typically, hardware MD algorithms use SAD (sum of absolute difference between original pixels and predicted pixels) [9] or SATD (sum of absolute transform difference) [10] in judging the best mode to alleviate the computational burden and avoid the reconstruction loop. However, the main problem of this kind of hardware MD is that the coding performance is much worse than that of the RDO-based method. This work proposes a novel low-complexity hardware-oriented MD scheme based on R-D estimation which also achieves high coding performance. The rest of this paper is organized as follows. Section 2 analyzes the challenges of hardware RDO-based MD. Section 3 detailedly presents the proposed hardware-oriented MD scheme. In section 4, experimental and implementation results are shown.

2. CHALLENGES ANALYSIS OF HARDWARE RDO-BASED MD

In hardware RDO-based MD design, many challenges need to be addressed and we list them as following:

1) *Block-level intra data dependency.*
2) *Bottleneck time of the processing units.*
3) *Highly efficient pipeline architecture.*
4) *Large hardware resource consumption.*

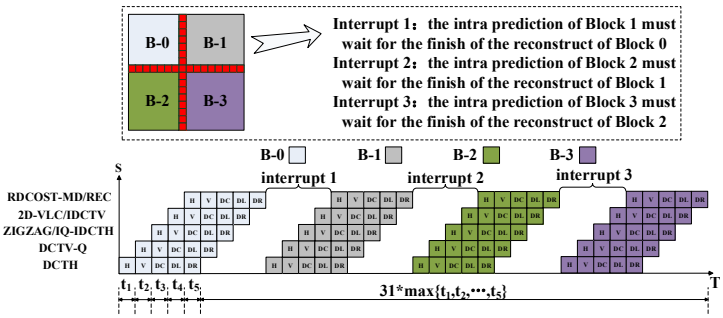

Fig. 2. Block-level data dependency in AVS.

The first challenge is the block-level intra data dependency which stems from the reconstruction loop caused by the intra prediction. In AVS, the data dependency comes from the block-level intra luma blocks, as shown in Figure 2, and the pipeline (generally, pipeline design is adopted in RDO-based MD to reduce the processing time) will be interrupted once every five modes to wait for the reconstruction of the proceeding block to finish. A 5-stage pipeline architecture is proposed by [11], as shown in the lower part of Figure 2, which shows many bubbles in the pipeline space time-diagram because of the data dependency.

The second challenge is the bottleneck (e.g. $\max\{t_1,t_2,\ldots,t_5\}$ in Figure 2) of the pipeline which will be a big drawback to increase the throughput (context-based 2D–VLC entropy coding unit generally be the bottleneck in AVS MD pipeline).

Furthermore, due to the above two challenges, how to design highly efficient pipeline architecture and save the corresponding hardware resource will be another two challenges.

To address the challenges talked above, we first propose a hardware-friendly *RDcost* estimation method and then present an REC-ORG united intra prediction scheme to break the data dependency in RDO-based MD. Finally, highly efficient MD pipeline architecture is put forward to enhance MD processing capacity.

To assist clarifying this work, the adopted five-stage MB level pipeline architecture of our encoder system is shown in Figure 3.

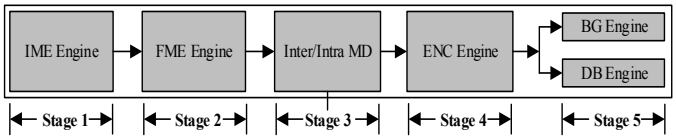

Fig. 3. Our adopted five-stage MB pipelining architecture.

The five stages of the MB level pipelining architecture are integer motion estimation (IME), fractional motion estimation (FME), Inter/Intra MD (MD), encoding engine (ENC), Bit stream generating engine (BG) and in-loop DeBlocking filter (DB). Among these stages, MD module chooses the best mode according to R-D value and ENC proceeds DPCM coding process for the best mode.

3. PROPOSED HARDWARE-ORIENTED MD SCHEME

3.1. *RDcost* estimation

In paper [8], the authors proposed a low complexity R-D estimation scheme, as shown in Figure 4, the estimated R and D can be directly calculated after quantization for each mode. However, the R-D model parameters should be updated adaptively which is hard to implement in hardware design. Paper [12] simplified the method in [8] and utilized it in discarding quantized coefficient process. In this paper, we propose a low complexity R-D estimation method for MD.

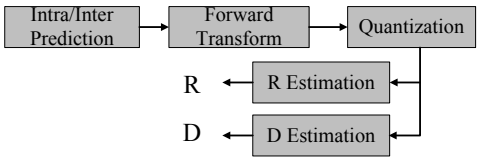

Fig. 4. R-D estimation scheme

About the estimated R (R_e), we first find out the solution "form" and then use the least squares technique to make roughly estimation on it. In paper [13], the authors modeled the probability density function (PDF) of the discrete cosine transform (DCT) coefficients with zero-mean generalized Gaussian distribution (GGD), as shown in Figure 5.

The GGD is described as follows:

$$P_{C_{uv}} = f_{uv}(C_{uv}) = \frac{\eta_{uv}\alpha_{uv}(\eta_{uv})}{2\sigma_{uv}\Gamma(1/\eta_{uv})}\exp\left\{-\left[\alpha_{uv}(\eta_{uv})\frac{|C_{uv}|}{\sigma_{uv}}\right]^{\eta_{uv}}\right\} \quad (1)$$

which

$$\alpha_{uv}(\eta_{uv}) = \sqrt{\frac{\Gamma(3/\eta_{uv})}{\Gamma(1/\eta_{uv})}} \quad (2)$$

where $\Gamma(\bullet)$ is the gamma function, η_{uv} and σ_{uv} are positive real-valued distribution parameters which control the shape and scale of the GGD, respectively. C_{uv} is transform coefficient, and $\hat{C}_{uv}$ denotes the quantized value of C_{uv}. The entropy coding bits of a quantized coefficient $\hat{C}_{uv}$ with occurrence probability $P_{\hat{C}uv}$ is directly dependent on the self-information [8] and we use it to estimate the coding bits r_{uv}. Self-information is defined by

$$r_{uv} = -\log_2(P_{\hat{C}uv}) \quad (3)$$

According to (3), to formulate r_{uv}, we first need to find out $P_{\hat{C}uv}$. Standards like H.264, AVS and HEVC, uniform scalar quantization method is used. Suppose the quantization step size Q_{step}, then we have

$$\hat{C}_{uv} = C_{uv} / Q_{step} \quad (4)$$

Then, corresponding to the quantized coefficient $\hat{C}_{uv}$, we use (5) to estimate the transform coefficient C_{uv}.

$$C_{uv} \approx \hat{C}_{uv} \bullet Q_{step} \quad (5)$$

Based on the theory of probability, for each Q_{step} interval of the horizontal axis in Figure 5, the probability is shown in the following (6).

$$P_{\hat{C}uv} \approx f_{uv}(\hat{C}_{uv} \bullet Q_{step}) \bullet Q_{step} \quad (6)$$

Combining (1), (3) and (6), the following (7) can be developed.

$$\begin{aligned}
r_{uv} &= -\log_2(P_{\hat{C}uv}) = -\log_2(\, f_{uv}(\hat{C}_{uv} \bullet Q_{step}) \bullet Q_{step}) \\
&= -\log_2\left\{\frac{\eta_{uv}\alpha_{uv}(\eta_{uv})Q_{step}}{2\sigma_{uv}\Gamma(1/\eta_{uv})}\exp\left\{-\left[\alpha_{uv}(\eta_{uv})\frac{|\hat{C}_{uv} \bullet Q_{step}|}{\sigma_{uv}}\right]^{\eta_{uv}}\right\}\right\} \\
&= -\left\{\log_2\left(\frac{\eta_{uv}\alpha_{uv}(\eta_{uv})Q_{step}}{2\sigma_{uv}\Gamma(1/\eta_{uv})}\right) + \log_2\left\{\exp\left\{-\left(\left[\alpha_{uv}(\eta_{uv})\frac{Q_{step}}{\sigma_{uv}}\right]^{\eta_{uv}}\right)\left|\hat{C}_{uv}\right|^{\eta_{uv}}\right\}\right\}\right\} \\
&= -\left\{\log_2\left(\frac{\eta_{uv}\alpha_{uv}(\eta_{uv})Q_{step}}{2\sigma_{uv}\Gamma(1/\eta_{uv})}\right) + \ln\left\{\exp\left\{-\left(\left[\alpha_{uv}(\eta_{uv})\frac{Q_{step}}{\sigma_{uv}}\right]^{\eta_{uv}}\right)\left|\hat{C}_{uv}\right|^{\eta_{uv}}\right\}\right\} / \ln 2\right\} \\
&= -\log_2\left(\frac{\eta_{uv}\alpha_{uv}(\eta_{uv})Q_{step}}{2\sigma_{uv}\Gamma(1/\eta_{uv})}\right) + \left(\left[\alpha_{uv}(\eta_{uv})\frac{Q_{step}}{\sigma_{uv}}\right]^{\eta_{uv}}\right)\left|\hat{C}_{uv}\right|^{\eta_{uv}} / \ln 2 \\
&= a_{uv}\left|\hat{C}_{uv}\right|^{\eta_{uv}} + b_{uv}
\end{aligned} \quad (7)$$

which

$$a_{uv} = \left[\alpha_{uv}(\eta_{uv})\frac{Q_{step}}{\sigma_{uv}}\right]^{\eta_{uv}} / \ln 2$$

$$b_{uv} = -\log_2\frac{\eta_{uv}\alpha_{uv}(\eta_{uv})Q_{step}}{2\sigma_{uv}\Gamma(1/\eta_{uv})}$$

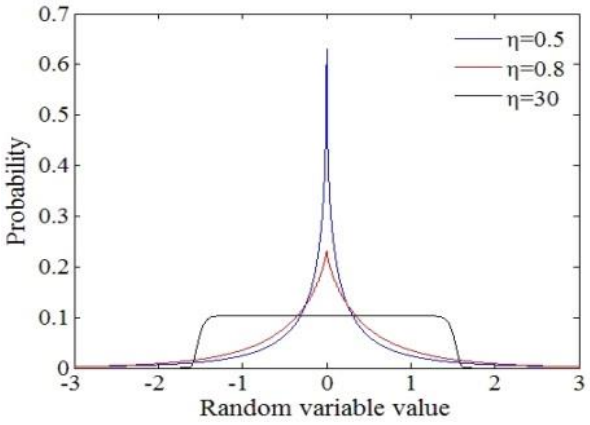

Fig. 5. Generalized Gaussian Distribution.

Then for one block (such as one 8×8 block), the estimated coding bits R_e will be

$$\begin{aligned}
R_e &= \sum_u\sum_v r_{uv} = \sum_u\sum_v\left(a_{uv}\left|\hat{C}uv\right|^{\eta_{uv}} + b_{uv}\right) \\
&= \sum_u\sum_v\left(a_{uv}\left|\hat{C}uv\right|^{\eta_{uv}}\right) + \sum_u\sum_v b_{uv} \\
&= \sum_u\sum_v\left(w_{uv}\left|\hat{C}uv\right|^{\eta_{uv}}\right) + \xi = X \bullet W
\end{aligned} \quad (8)$$

which

$$\begin{aligned}
&\xi = \sum_u\sum_v b_{uv} \\
&w_{uv} = a_{uv} \\
&X = \left[\left|\hat{C}_{00}\right|^{\eta}, \left|\hat{C}_{01}\right|^{\eta}, \ldots, \left|\hat{C}_{77}\right|^{\eta}, 1\right] \\
&W = [w_{00}, w_{01}, \ldots\ldots, w_{77}, \xi]^{\mathrm{T}}
\end{aligned} \quad (9)$$

We will use (8) to produce the estimation of the coding bits R_e for an 8×8 coding unit. To compute R_e, we pre-fit the weighting matrix W by using least squares method. The pre-fitting process is shown in the following.

According (8) and (9), we know,

$$\begin{bmatrix} X_1 \\ X_2 \\ \ldots \\ X_n \end{bmatrix} \bullet W = \begin{bmatrix} |x_{100}|^{\eta}, |x_{101}|^{\eta}, \ldots, |x_{177}|^{\eta}, 1 \\ |x_{200}|^{\eta}, |x_{201}|^{\eta}, \ldots, |x_{277}|^{\eta}, 1 \\ \ldots \\ |x_{n00}|^{\eta}, |x_{n01}|^{\eta}, \ldots, |x_{n77}|^{\eta}, 1 \end{bmatrix} \bullet W = \begin{bmatrix} R_1 \\ R_2 \\ \ldots \\ R_n \end{bmatrix} \quad (10)$$

where R_i represents the real coding bits of i^{th} coding unit. From (13), we can get W as the following (11):

$$W = \left(X^{\mathrm{T}} \bullet X\right)^{-1} X^{\mathrm{T}} \bullet R \quad (11)$$

In (11), the superscript "-1" denotes the inverse of a matrix. Based on (11), we develop the weighting matrix W.

To make more accurate estimation, we use different weighting matrix W for intra and inter modes. The elements in W are floating-point values, and we multiply them by 256 to make them fix-point values ($\hat{w}$) which are more suitable for hardware platform (FPGA, ASIC, etc.). Taking W_{intra} for example, we depict the fix-point generating process as Figure 6.

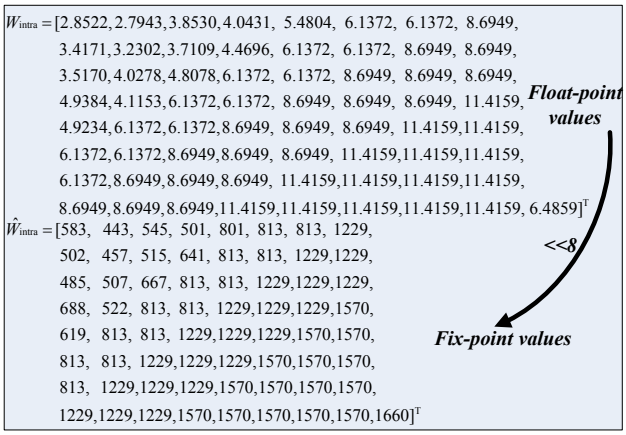

Fig. 6. Fix-point values of weighting matrix generation for hardware implementation.

In paper [13], the authors pointed out that most of natural images were best fitted with shape parameter η=0.5 and the value η for the different coefficients does not vary much within a single image. We choose η=0.5 for simplification.

Finally, we illustrate the R estimation process by taking a 4×4 block for example (here we use 4×4 block only for the convenience), as shown in Figure 7.

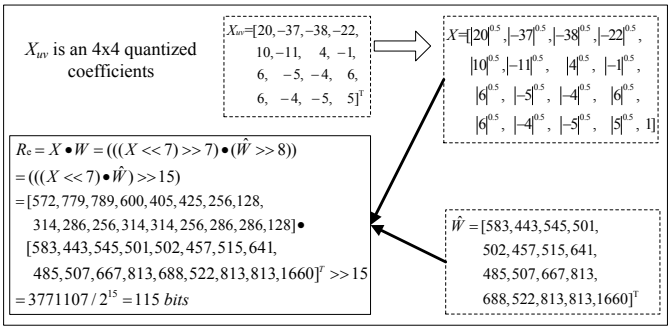

Fig. 7. R estimation process for one 4×4 quantized block.

Besides, we use look-up tables, which store the pre-calculated enlarged square roots of quantized coefficients (to save hardware resources, we treat all quantized coefficients larger than 512 as 512), to avoid the usage of multiplier in hardware design.

For the distortion estimation (D_e), as work [8] tells that for a quantized coefficient $\hat{C}_{uv}$, the distortion d_{uv} can be calculated in the transform domain after quantization, as shown in the following.

$$D_e = \sum_u \sum_v d_{uv} \tag{12}$$

which

$$d_{uv} = \left\lfloor \left| offset_{uv} - low\_qbits_{uv} \right| / 2^{q\_bits} \bullet Q_{step} \right\rfloor^2 \tag{13}$$

where $offset_{uv}$ is quantization rounding offset and low_qbits_{uv} is the discarded low q bits in quantization process [8]. We use (13) as estimated distortion for a quantized block.

Based on R_e and D_e, we can get the estimated $RDcost_e$ according to (14) for each mode and then choose the best modes.

$$RDcost_e = D_e + \lambda R_e \tag{14}$$

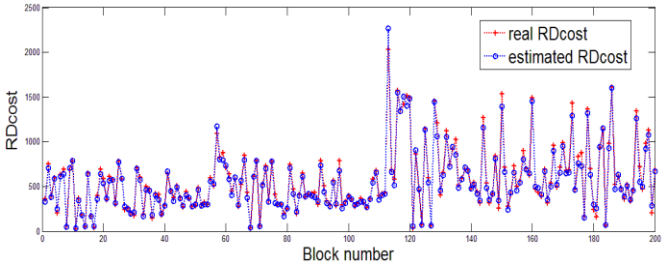

Fig. 8. Real *RDcost* and estimated *RDcost* of randomly selected 200 blocks.

We can see from Figure 8 that the proposed *RDcost* estimation algorithm is accurate, and the estimated *RDcost*s are closely matched with the actual *RDcost*s for both low-bits and high-bits blocks.

3.2. Proposed REC-ORG united intra prediction scheme

To break data dependency, we propose a REC-ORG united intra prediction scheme: using original (ORG) pixels instead of the reconstructed (REC) pixels for the red ones in Figure 9 and using reconstructed pixels for the green ones in Figure 9 to proceed intra prediction.

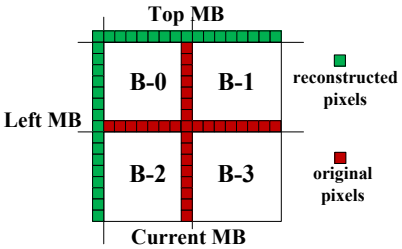

Fig. 9. REC-ORG united intra prediction.

However, according to Figure 2, the REC pixels of the left MB in Figure 9 are just available after the processing of ENC stage. In traditional raster coding order, as shown in Figure 10, when MD is processing the n^{th} MB, the ENC stage is processing the $(n\text{-}1)^{th}$ MB and thus the left sixteen green reconstructed pixels in Figure 9 are always not available for MD of the n^{th} MB.

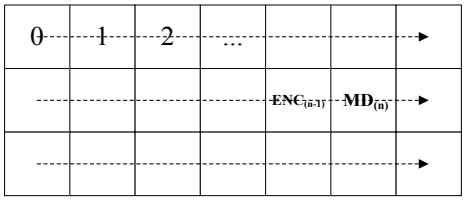

Fig. 10. Traditional raster coding order.

To make the left sixteen pixels available, we change the MB-level coding order from traditional raster order to zigzag coding order in Figure 11. As depicted in Figure 11, the top and the left reconstructed pixels are both available because the top and the left MBs have already been encoded.

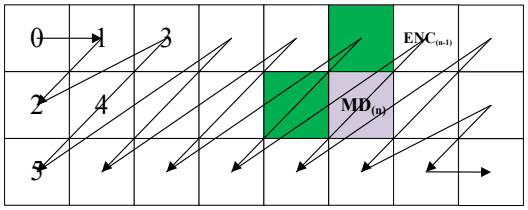

Fig. 11. Zigzag coding order.

By REC-ORG united intra prediction scheme, our MD can not only break the data dependency but also maintain high coding performance.

3.3. Proposed highly efficient MD pipeline architecture

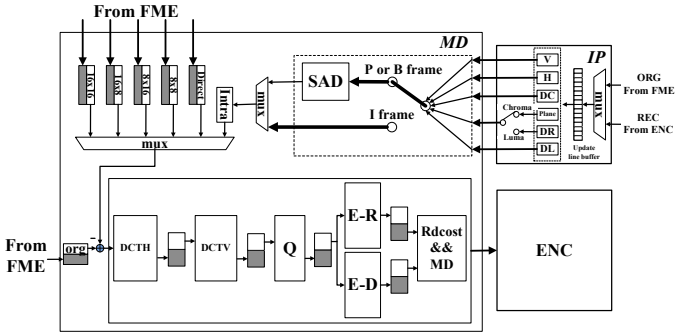

Fig. 12. Our proposed MD architecture.

We take AVS for example to illuminate the proposed highly efficient MD pipeline architecture. As for H.264, similar architecture can be developed. Our proposed MD architecture is shown in Figure 12, and the R-D estimation core (lower part in Figure 12) is divided into 5 stages and the first to third stages are horizontal DCT (DCTH), vertical DCT (DCTV) and Q. After quantization, the pipeline is divided into 2 branches. Rate estimation (E-R) and distortion estimation (E-D) are both belong to the fourth stage, and they again merge into one part at the fifth stage: COST-MD. COST-MD calculates R-D cost and chooses the best mode. All the buffers between 2 consecutive stages are ping-pong mode buffers. In this work, we adopt 8-pixel parallelism (processing 8 pixels in each cycle) and the time consumptions for each stage are tabulated as Table 1.

Table 1. Time consumption of basic processing units.

Processing Unit (8-pixel parallelism)		Time Consumption (cycles)	
DCTH		17	
DCTV		18	
Q		16	
E-R	E-D	18	18
COST-MD		14	

In this work, for intra block of I frame, we choose the best luma modes from {V, H, DC, DR and DL} and choose the best chroma modes from {V, H, DC, P} according to R-D value. We choose the best modes from {Pskip, P16×16, P16×8, P8×16, P8×8, Intra8×8} for P frame and the best modes from {Bdirect, Bskip, B16×16, B16×8, B8×16, B8×8, Intra8×8} for B frame also based on R-D value. For P and B frames, we use SAD to decide the best block-level modes of Intra8×8.

For Pskip and Bskip modes, reconstructed pixels are identical to the predicted pixels, thus real D can be yield directly with original pixels and predicted pixels. Besides, the real coding bits R equals to 0. Therefore, Pskip and Bskip modes need not be processed by R-D estimation core, and the pipeline space time-diagram of I, P and B frames can be depicted as Figure 13-15.

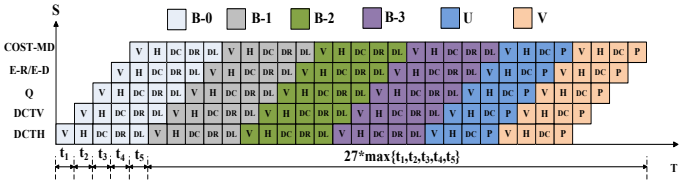

Fig. 13. Pipeline space time-diagram for I-frame.

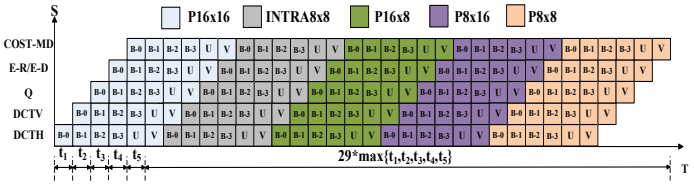

Fig. 14. Pipeline space time-diagram for P-frame.

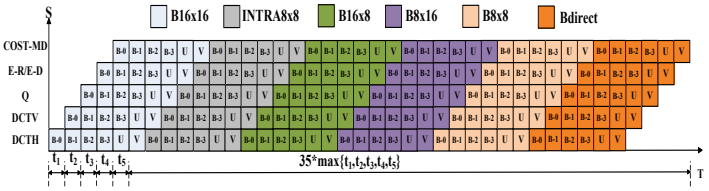

Fig. 15. Pipeline space time-diagram for B-frame.

Combining Table 1 and Figure 13-15, we can roughly estimate the processing time of MD for I, P and B frames as the following (15) to (17).

$$T_I = \sum_{i=1}^{5} t_i + 27 \times \max\{t_1, t_2, \ldots, t_5\} = 83 + 27 \times 18 = 569 \ cycles \quad (15)$$

$$T_P = \sum_{i=1}^{5} t_i + 29 \times \max\{t_1, t_2, \ldots, t_5\} = 83 + 29 \times 18 = 605 \ cycles \quad (16)$$

$$T_B = \sum_{i=1}^{5} t_i + 35 \times \max\{t_1, t_2, \ldots, t_5\} = 83 + 35 \times 18 = 713 \ cycles \quad (17)$$

From Figure 13-15, we can also see that the bubbles in the pipeline space time-diagram of Figure 2 disappear. Besides, the general bottleneck (entropy coding unit) of RDO-based MD pipeline architecture is avoided and thus $\max\{t_1, t_2, \ldots, t_5\}$ decreases. Therefore, the pipeline throughput increases when compared with [11].

4. EXPERIMENTATAL RESULTS AND IMPELMENTATION

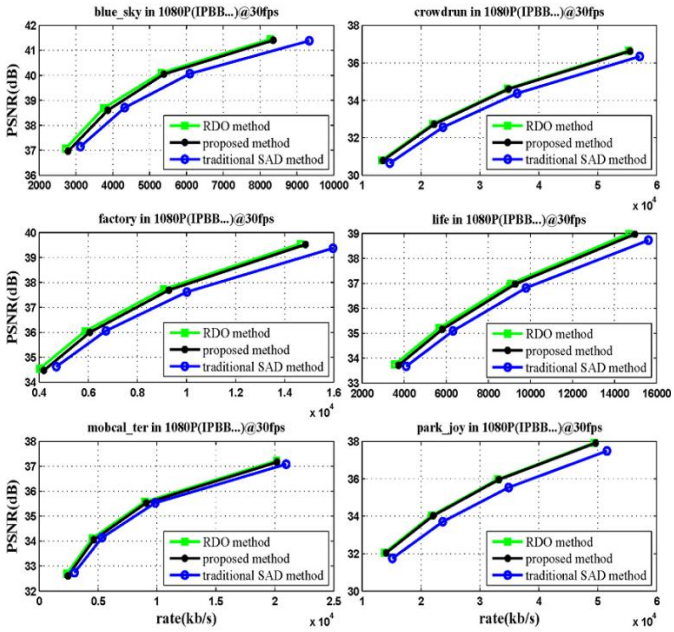

Fig. 16. Performance comparisons of different mode decision algorithms for sequences with 1080P format.

Table 2. Comparisons for different sequences.

Format	Sequence	Proposed method VS TRUE RDO		Proposed method VS SAD [9]	
		PSNR Loss (dB)	Bit-rate change (%)	PSNR Gain (dB)	Bit-rate change (%)
1080P	blue sky	0.144	3.867	0.409	-9.593
	crowdrun	0.055	1.329	0.419	-9.574
	factory	0.112	3.399	0.376	-8.864
	life	0.126	3.459	0.369	-9.232
	mobcal	0.093	4.468	0.205	-9.093
	park joy	0.059	1.273	0.639	-12.77

Different sequences under different quantization parameters were tested and Figure 16 shows the RD curves of the selected 6 sequences. Table 2 shows the tabulated performance comparison of the proposed algorithm with true RDO and traditional SAD method [9]. From Table 2 we can see that our proposed method achieves similar performance to true RDO method (0.098dB PSNR loss in average), and far outperforms the traditional SAD method (0.402dB PSNR gain in average).

Our proposed MD architecture is implemented by Verilog-HDL language and verified on Virtex5 FPGA LX330. The slice consumption on LX330 is 18%. With our proposed architecture and taking all the overheads cycles into account, our MD unit can accomplish one MB-level MD in 606 cycles, 642 cycles and 750 cycles for I, P and B frames, respectively. The processing capacity is increased by 29%, 23% and 23% for I, P, B frame when compared with work [11].

5. REFERENCES

[1] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," IEEE Signal Process. Mag., vol. 15, no. 6, pp. 74–90, Nov. 1998.

[2] AVS Video Expert Group, Information Technology — Advanced Audio Video Coding Standard Part 2: Video, in Audio Video Coding Standard Group ofChina (AVS), Doe. AVS-N1063, Dec. 2003.

[3] Wiegand, T., Sullivan, G., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC Video Coding Standard. IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560 – 576, Jun. 2003.

[4] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1648–1667, Dec. 2012.

[5] F. Pan, X. Lin, R. Susanto, K. P. Lim, Z. G. Li, G. N. Feng, D. J. Wu, and S. Wu, "Fast mode decision algorithm for intra prediction in H.264/AVC video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 7, pp. 813–822, Jul. 2005.

[6] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, R. Susanto, and C. C. Kuo, "Fast intermode decision in H.264/AVC video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 6, pp. 953–958, Jul. 2005.

[7] Y.-K. Tu, J.-F. Yang, and M.-T. Sun, "Efficient rate-distortion estimation for H264/AVC coders," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 5, pp. 600–611, May 2006.

[8] Xin, Z., Sun Jun, Siwei Ma, Wen Gao., "Novel Statistical Modeling, Analysis and Implementation of Rate-Distortion Estimation for H.264/AVC Coders," Circuits and Systems for Video Technology, IEEE Transactions on, 2010. 20(5): p. 647-660.

[9] K. Babionitakis, G. Doumenis, G. Georgakarakos, G. Lentaris, K. Nakos, D. Reisis, I. Sifnaios and N.Vlassopoulos, "A real-time H.264-AVC VLSI encoder architecture," Journal of Real-Time Image Processing , vol. 3, Numbers 1-2, pp. 43-59, 2008.

[10] Chun-Wei Ku, Chao-Chung Cheng, Guo-Shiuan Yu, Min-Chi Tsai, Tian-Sheuan Chang, " A High-Definition H.264/AVC Intra-Frame Codec IP for Digital Video and Still Camera Applications," Trans. Circuits Syst. Video Technol., vol. 16, no. 8, pp. 917-928, Aug. 2006.

[11] Chuang Zhu, Yuan Li, Hui-zhu Jia, Xiao-dong Xie, Hai-bing Yin, "A highly efficient pipeline architecture of RDO-based mode decision design for AVS HD video encoder ," ICME, July 2011, 2011.

[12] Chuang Zhu, Huizhu Jia, Jie Liu, Xianghu Ji, Hao Lv, Xiaodong Xie and Wen Gao, "Multi-Level Low-Complexity Coefficient Discarding Scheme for Video Encoder," ISCAS2014，accepted.

[13] F. Müller, "Distribution shape of two-dimensional DCT coefficients of natural images," Electron. Lett., vol. 29, no. 22, pp. 1935–1936, Oct. 1993.