# Coding Unit and Prediction Unit Pre-selection in HEVC Intra Encoding

Xianyu Yu, Guodong Liu, Jia Zhu, Zhenyu Liu, Dongsheng Wang*
IMETU and TNList, Tsinghua University
Beijing 100084, China.

**Abstract— HEVC doubles the coding efficiency with more than 4x coding complexity as compared to H.264/AVC. To alleviate the burden of Intra encoder, we estimate the RD-cost from the source image textures, and dynamically select two promising CU/PU mode candidates to execute exhaustive RDO processing. As integrated in our hardwired encoder, the averaged 61.7% computation complexity was saved with 4.53% rate augment. With TSMC 90nm technology, the real-time encoder for HDTV1080p at 44fps is implemented with 2269k-gate at 357MHz operating frequency.**

## I. INTRODUCTION

High Efficiency Video Coding (HEVC)[1] is the state-of-the-art video compression standard developed by Joint Collaborative Team on Video Coding (JCT-VC). HEVC aims to fulfill the growing requirements for higher quality and resolutions in video devices and applications, which are beyond the capabilities of H.264/AVC [2]. Thus, HEVC is devised to save around 35-40% bit-rate cost compared to H.264/AVC High Profile, while providing the equivalent object video quality[3].

To handle the large picture size, HEVC provides the flexible quad-tree structure based coding tree unit (CTU), which is composed of basic coding unit (CU), prediction unit (PU), and transform unit (TU). The $2N \times 2N$ ($N \in \{8, 16, 32\}$) CTU is the root of coding tree, which can be further split to four $N \times N$ CU, and this splitting procedure is feasible for each CU when its size is greater than $8 \times 8$. In the Intra prediction, PU size is equal to its $2N \times 2N$ CU when $N \geq 8$. On the other hand, $8 \times 8$ CU can use $8 \times 8$ or $4 \times 4$ PU sizes. The residue block of CU is further partitioned with a quad-tree structure, which is denoted as residue quad-tree (RQT), and the transform is processed on each leaf node, i.e., TU.

For each candidate parameter vector $\vec{p}$ (including CU/PU/TU structures), its coding performance is evaluated by the Lagrangian multiplier optimization technique,

$$J(\vec{p}) = D(\vec{p}) + \lambda \cdot R(\vec{p}), \tag{1}$$

in which, $D(\vec{p})$ and $R(\vec{p})$ represent the distortion and rate costs, respectively, and $\lambda$ is the Lagrangian multiplier controlling the rate-distortion trade-off. To derive the accurate values of $D(\vec{p})$ and $R(\vec{p})$, the encoder must carry out the time consuming transform, quantization, inverse quantization, inverse transform, and entropy coding procedures. The massive CU/PU/TU mode configurations lead to intensive computation of HEVC encoder. In HM reference software, the encoder exhaustively traverses all parameter vectors to find the best candidate

$$\vec{p}_o = \arg\min_{\vec{p}} \{D(\vec{p}) + \lambda \cdot R(\vec{p})\}. \tag{2}$$

Experiments reveal that the coding time increase monotonically with the CU depth number.

To overcome the aforementioned obstacle, plethora algorithms have been proposed for fast intra coding, which fall into three primary categories: The first one substitutes the full RD-cost calculation with the low-complexity RD-cost estimation[4, 5]; The methods of the second kind rely on filtering out the most impossible candidate CU[6] or PU[7] modes; While, the third class of methods speed up the searching by early terminating the RDO execution during CU[8, 9], PU[10], or TU[11] search. However, the above fast algorithms are based on the software oriented recursive CU-depth search processing. As integrated in the real-time hardwired encoder systems, the CU level parallelism either degrades the performance or introduces considerable hardware overhead. In addition, the speedup efficiency of the early termination and the historical statistics based dynamic CU depth determination methods depends on the statistic features of the videos. Namely, the above fast algorithms could not ensure a stable speedup performance, which is essential for the real-time encoding system.

In this paper, we devise the VLSI friendly CU/PU mode decision algorithm for Intra prediction in HEVC hardwired encoder design. The proposed algorithm is primarily composed of the following three steps: First, the edge strength and direction of each $N \times N$ ($N \in 4, 8, 16, 32$) partitions in CTU are derived. Secondly, the linear model between predict error matrix of $N \times N$ partition and its edge strength matrix is constructed by using weighted least square linear regression. It should be noticed that the model is obtained by off-line learning. From the estimated prediction error, we can evaluate the RD-cost of $N \times N$ partition. Thirdly, our algorithm dynamically chooses one candidate for RDO from $16 \times 16$ and $32 \times 32$ CU modes. $8 \times 8$ CU mode RDO is always executed in our proposal. However, we dynamically determine its PU mode ($8 \times 8$ or $4$) according to their estimated RD-cost comparisons. We always discard $64 \times 64$ CU mode, which merely saves less than 0.22% rate in HD1080p and WQXGA ($2560 \times 1600$) sequences. The hardwired Intra encoder is implemented integrated with the proposed fast algorithm and its performance is demonstrated.

The rest of this paper is organized as follows. In Section II, a brief introduction of our HEVC Intra encoding flow is de-

*This work is funded by TNList cross-discipline foundation,the Nature Science Foundation of China (Grant No.60833004 and 60902101), and the National 863 High-Tech Programs of China (No.2012AA010905).

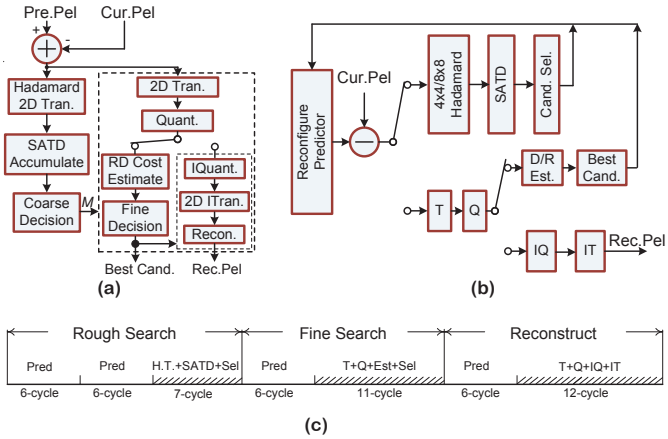



Fig. 1. HEVC prediction mode decisions for a $N \times N$ CU. (a)Flowchart for software. (b)Design for hardware. (c)Processing schedule of $4 \times 4$ CU. (Pred: Prediction signal generation; H.T.: Hadamard Transform; SATD: SATD computation; Sel: Select candidate modes; T: Transform; Q: Quantization; Est: RD-cost estimation; IQ: Inverse quantization; IT: Inverse transform)

scribed. In Section III, the hardware friendly mode decision algorithm is presented, including the theoretical analysis, parameters study method, and RD-cost estimation schemes. Our hardware design and experimental results are illustrated in Section IV , followed by the conclusions in Section V.

## II. Overviews of HEVC Intra Coding

Carrying forward the advantages of H.264/AVC intra coding framework, HEVC standard also employs the block-based hybrid coding architecture, which is developed on the spatial prediction, followed by the transform coding and post processing steps. To handle the HDTV and UHDTV video sizes, HEVC brings in some advanced techniques contributing to the compression efficiency, including the quad-tree based coding unit structure, fine angular directional prediction, 16-bit length variable-block-size DCT/DST transforms, and prediction direction-based transform coefficient scanning.

The quad-tree coding block partition structure efficiently utilizes variable sizes of Coding, Prediction and Transform Units(CU/PU/TUs). As mentioned in Section I, RDO must be exhaustively executed to select the best coding parameters, i.e., CU quad-tree structure, PU and RQT partitions in each CU. In addition, to enhance the spatial prediction accuracy, HEVC employs 33 directional predictions as well as Planar and DC in PU mode search. The numerous Intra prediction modes impose a heavy burden to the encoder. The HM reference software employs the fast heuristic Intra prediction mode selection algorithm, which is composed of the low-complexity rough mode decision and the full RDO based fine mode search. Specifically, the rough decision select several candidates with the minimum SATD based RD-costs out of all 35 prediction modes. Fine mode decision then choose the best Intra mode through the full RDO.

Even with the rough to fine heuristic RDO, the associated hardware and timing costs are both unfeasible for HDTV1080p real-time encoder implementations. The primary hindrance still comes from the full RDO. For the hardware consideration, even one variable block size DCT/DST accelerator consumes 320k-gate stand cells, let alone multiple engines required by parallelism RDO. On the other hand, the rate cost estimation in HM reference software is the CABAC alike algorithm, which is composed of the binarization and the bin-grain arithmetic coding. Even with the advanced ASIC accelerator, for one $32 \times 32$ block, the averaged 200-300 cycles are required in rate evaluation.

Our design simplifies the RD-cost estimation algorithm. Similar to literature [5], our design also obtains the distortion from quantization results. However, the original method merely saves the hardware of inverse transform and inverse quantization. Our contributions in the distortion evaluation algorithm include two aspects: First, we substitute the $N \times N$ DCT with the hardware saving $N \times N$ Hadamard counterparts as $N \geq 8$; Second, we restrict the bit-width of multipliers for SSE computation to 8-bit for saving chip area. By using binary classification and linear regression method, we develop the fast rate models, which are applied to estimate the rate according to the features of $N \times N$ quantization coefficient block. The latency of rate estimation is reduced to 3 cycles. In our proposal, the TU size is always equal to PU size, which introduces negligible coding quality loss. The flowchart of our PU mode decision algorithm and the corresponding system block are presented in Fig. 1(a) and (b), respectively. Figure. 1(c) provides the processing schedule of Luma $4 \times 4$ PU mode decision. Experiments illustrate that, on average, 1.27% rate increase is introduced by our simplified RDO.

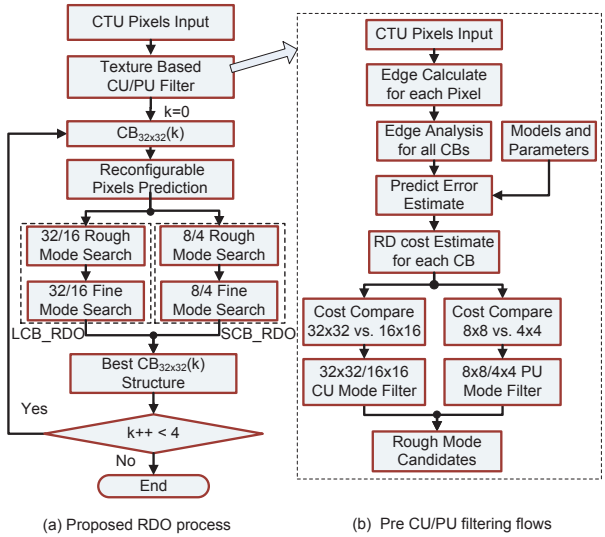
In the VLSI implementation, parallel processing is imperative for the HDTV1080p@30fps real-time encoding. In detail, the processing latency of RDO for $8 \times 8$ CU with $4 \times 4$ PU mode is 248 clocks (220 for 4 $4 \times 4$ Luma + 28 for 1 $4 \times 4$ Chroma). Even for $4 \times 4$ PU mode RDO, 241MHz operating frequency is required. The serial processing of all CU modes needs 577MHz clock speed. On the other hand, with 90nm CMOS technology and traditional ASIC design flow, the typical clock speed is around 400MHz (our design is 357MHz), falling far behind the requirement.

The primitive parallel architecture, in which each PU mode search is equipped with the dedicated process engine, will consume 5282k-gates. To reduce the chip area in parallel Intra HEVC encoder, we discard the redundant CU/PU mode candidates during RDO by investigating the source image textures. Figure 2 shows the flowchart of our parallel RDO engine embedded with the proposed CU/PU mode filtering algorithm. The principle and detailed analysis of our CU/PU mode reduction will be explained in Section III. It should be noticed that, we devise the reconfigurable prediction accelerator, which is shared by all CU/PU RDO processing. The VLSI architecture and processing schedule will be described in Section IV.

## III. CU/PU Candidate Reduction using RD-cost estimation from Source Image Texture Investigation

### A. Panoramic View of Proposed Method

As showed in Fig. 2, before the traditional coding configuration RDO, we first investigate the textures of source image




(a) Proposed RDO process (b) Pre CU/PU filtering flows

Fig. 2. Propoed parallel RDO flowchart for a luma CTU sized $64 \times 64$, using edge based pre CU mode filtering.

and filter out the improper CU/PU candidates. This module is denoted as "Texture Based CU/PU Filter" in Fig. 2. As mentioned in Section II, $64 \times 64$ CU is always avoided, so the remaining CU candidates include $32 \times 32$, $16 \times 16$, and $8 \times 8$. For each $32 \times 32$ coding block ($CB_{32\times32}$), we dynamically discard $32 \times 32$ or $16 \times 16$ CU mode RDO from the texture investigation. $8 \times 8$ CU possesses two PU mode candidates, i.e., $8 \times 8$ and $4 \times 4$. The dedicated PU mode for each $8 \times 8$ CU is also defined during the texture preprocessing stage. In the following RDO processing, two modules (LCB_RDO and SCB_RDO) are employed to work in parallel. LCB_RDO is in charge of deriving the RD-cost of current $32 \times 32$ CB with the predefined $32 \times 32$ or $16 \times 16$ mode. Simultaneously, SCB_RDO obtains all RD-costs of $8 \times 8$ partitions with the specified PU mode. From the results of LCB_RDO and SCB_RDO, we then assemble the best coding tree structure.

The detailed flowchart of texture analysis is illustrated in Fig. 2(b). As the CTU source image is given, we can calculate the edge strength and edge direction on every pixel. The edge statistic features, such as the main edge direction, edge strength, and edge direction distributions, in any CU are obtained. According to the edge feature classification, we develop the linear relations between the pixel edge strength and its prediction error power, which will be unravel in Section B. Therefrom, we construct the RD-cost estimation method (as shown in Section C) to discard the improper CU/PU candidate modes. As our method is based on the source image investigation, and eliminates the constant computation burden, it neither degrades the pipeline performance nor incurs the encoding complexity oscillation, which are prohibited in the real-time encoding system.

### *B. Linear Relation between Texture and Prediction Error*

Prediction error is the primary factor that determines the coding cost. Generally, the more accurate the prediction is, the less the RD cost will be. The prediction error power at position $(i, j)$ in a $N \times N$ CB is defined as

$$PE_k = (P_k - C_k)^2 \tag{3}$$

where, $i$ and $j$ denotes the ordinate and abscissa of the pixel (left up corner is (0,0)), $k = i \cdot N + j$, $P_k$ is the prediction pixel, and $C_k$ is the source pixel. As we know, generating and searching the best prediction signal $P_k$ require the intensive computation. Then, we strive for developing the model to fast estimate the power of $PE_k$.

The angular predictions in HEVC is on the basis of exploiting the local edge continuity. As expected, strong correlations exist between $PE_k$ and its edge strength $ES$. Except for the texture features, $PE_k$ is affected by the quantization noise $QE$. This is mainly because that the prediction signal $P_k$ is deduced from the decoded pixels, which have been polluted by the quantization. From the above analysis, we suppose that the prediction error power ($PE_k$) stems from $QE$ and $ES$, expressed as

$$PE_k \approx a \cdot Qs^2 + b_k \cdot ES_k. \tag{4}$$

where $a$ and $b_k$ are linear regression parameters. The term $a \cdot Qs^2$ indicates that the quantization error power $QE$ is linear with the square of quantization step ($Qs^2$).

$$\begin{aligned} &Qs = 2^{Qp/6} \cdot Q[Qp\%6] \\ &Q[i] \in \{0.625, 0.7031, 0.7969, 0.8906, 1, 1.125\} \end{aligned} \tag{5}$$

in which, $Qp$ is the quantization parameter, $/$ and $\%$ are the quotient and remainder operators, respectively. $ES_k$ at position $(i, j)$ is proportional to the edge gradient, written as

$$ES_k = eh_k^2 + ev_k^2, \tag{6}$$

where $eh_k$ and $ev_k$ denote the horizontal and vertical edge gradient, respectively.

For $N \times N$ CB, $N^2 + 1$ parameters, i.e., $a$ and $b_k$ ($k \in [0, N^2 - 1]$), are required. Through measurements, we can derive $M$ sample groups, including the edge strength on each pixel $ES_k(\tau)$, the quantization error $QE(\tau)$, and the actual prediction error power $PE_k(\tau)$, with $\tau \in [0, M-1]$. Let $\widetilde{PE}_k(\tau)$ represent the estimated prediction error power. Our target is that each prediction entry $\widetilde{PE}_k(\tau)$ approaches its actual value $PE_k(\tau)$ and the summary of $\widetilde{PE}_k(\tau)$ ($\sum_{k=1}^{N^2-1} \widetilde{PE}_k(\tau)$) also approaches the value ($\sum_{k=1}^{N^2-1} PE_k(\tau)$). Then, we have $N^2+1$ target functions. According to the weighted least squares estimation theory, the above question can be summarized as

$$\begin{aligned} \arg \min_{\{a,b_k\}} \Bigg\{ \sum_{\tau=0}^{M-1} \Bigg[ \sum_{k=1}^{N^2-1} w_k \cdot (PE_k(\tau) - \widetilde{PE}_k(\tau))^2 \\ + w_{N^2} \cdot \Big( \sum_{k=1}^{N^2-1} PE_k(\tau) - \sum_{k=1}^{N^2-1} \widetilde{PE}_k(\tau) \Big)^2 \Bigg] \Bigg\} \end{aligned} . \tag{7}$$

Let assume that estimation error $\varepsilon_k = PE_k - \widetilde{PE}_k$ are uncorrelated with each other, and $\varepsilon_k$ is a random variable with zero mean and constant variance(not changed with $k$). The optimum weighing vector $\vec{w}$ is

$$\begin{cases} w_k = 1 & 0 \le k \le N^2 - 1 \\ w_{N^2} = \dfrac{1}{N^2} \end{cases} . \tag{8}$$

The solution $\overrightarrow{\theta} = (a, b_0, \cdots, b_{N^2-1})^T$ of (7) is formulated as

$$\overrightarrow{\theta} = (\boldsymbol{A^T W A})^{-1} \cdot \boldsymbol{A^T W} \cdot \overrightarrow{PE}, \tag{9}$$

in which, $\boldsymbol{A}$ is composed of $M$ groups of measured $Qs^2$ and edge strengths, written as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}(0) \\ \boldsymbol{A}(1) \\ \vdots \\ \boldsymbol{A}(M-1) \end{bmatrix}. \tag{10}$$

Each sub-matrix $\boldsymbol{A(\tau)}$ is defined as

$$\boldsymbol{A}(\tau) = \begin{bmatrix} Qs^2(\tau) & ES_0(\tau) & 0 & \cdots & 0 \\ Qs^2(\tau) & 0 & ES_1(\tau) & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & \vdots \\ Qs^2(\tau) & 0 & 0 & \cdots & ES_{N^2-1}(\tau) \\ N^2 \cdot Qs^2(\tau) & ES_0(\tau) & ES_1(\tau) & \cdots & ES_{N^2-1}(\tau) \end{bmatrix}. \tag{11}$$

The vector $\overrightarrow{PE}$ denominates the sampled prediction errors

$$\overrightarrow{PE} = \begin{bmatrix} \overrightarrow{PE}(0) \\ \overrightarrow{PE}(1) \\ \vdots \\ \overrightarrow{PE}(M-1) \end{bmatrix}, \tag{12}$$

in which, each sub-vector is composed of all prediction error entries and their sum in one $N \times N$ CB .

$$\overrightarrow{PE}(\tau) = \begin{bmatrix} PE_0(\tau) \\ PE_1(\tau) \\ \vdots \\ PE_{N^2-1}(\tau) \\ \sum_{k=0}^{N^2-1} PE_k(\tau) \end{bmatrix}, \tag{13}$$

$\boldsymbol{W}$ is the diagonal weighting matrix having the form of

$$\boldsymbol{W} = \mathrm{diag}(\boldsymbol{W}(0), \boldsymbol{W}(1), \cdots, \boldsymbol{W}(M-1)), \tag{14}$$

and each sub-matrix $\boldsymbol{W}(\tau)$ is also a diagonal matrix $\boldsymbol{W}(\tau) = \mathrm{diag}(1, 1, \cdots, 1, 1/N^2)$.

To further improve the accuracy of our estimation algorithm, the sampled CBs are classified according to their texture distribution features, and each class possesses the dedicated model. As literature [12], for each CB, we can derive its edge direction histogram from its edge mapping. The edge direction histogram is composed of 33 cells corresponding to 33 prediction directions. From the distribution of histogram cells, the CB is classified with respect to edge direction homogeneity, prominent angle direction, and prominent angle strength.

**Prominent angle direction** In the histogram, the cell with the maximum amplitude points out the main angle direction, which is divided into 4 categories. The 33 prediction angles are indexed from 2 to 34 in HEVC. The first category (D0) ranges from 7 to 13, represents the horizontal alike directions; The second category (D1) includes modes from 23 to 29, represents the vertical alike directions; The third one (D2) includes -45 degree alike directions (modes 14 to 22); The other directions make up the fourth category (D3).

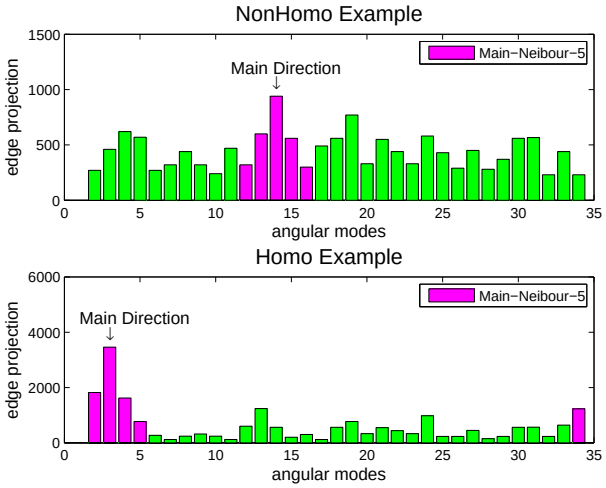



Fig. 3. Histogram of directional homogeneity examples

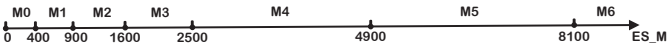



Fig. 4. Seven classes ($M0$ to $M6$) distinguished with the maximum edge strength.

**Directional homogeneity** If the edges in a CB are concentrated in one direction, this CB is denoted as directional homogeneous; Otherwise, it is nonhomogeneous. In detail, let $\sigma$ denote the sum of histogram cells of main direction and its 4 neighbors, $\Sigma$ is the sum of all histogram cells, if $\sigma/\Sigma > 1.0 - 0.1 \log_2 N$, the $N \times N$ CB is directional homogeneous. Otherwise, it is labeled as nonhomogeneous. Figure 3 illustrates the homogeneous and nonhomogeneous examples.

**Prominent angle strength** Even though prediction error always increases monotonically with the edge amplitude, but the relationship is not linear. Therefore, we empirically develop 7 groups according to the prominent angle strength, which is depicted as Fig.4.

As employing the classification strategies, each $N \times N$ ($N \in \{4, 8, 16, 32\}$) CB has totally 56 linear model candidates. Parameters $\overrightarrow{\theta}$ are studied specially with respect to the CB size and the texture classifications. The parameters $b_k$ of $8 \times 8$ CB with directional homogeneity and angle strength M0 are demonstrated by Fig.5. It can be observed parameter values monotonically increase with the ordinate and abscissa. This is because the degradation of prediction performance with the distance away from the boundary references. Another prominent feature is that the parameter values in D3 are much larger than those in other directional groups. With the classification based linear regression, the prediction performance is improved by the averaged 23%.

### *C. Prediction Error based RD Cost Estimation*

The ultimate target is to derive the RD-cost estimation. Once the value of $\widetilde{PE}_k$ is obtained, we can evaluate the correspond-

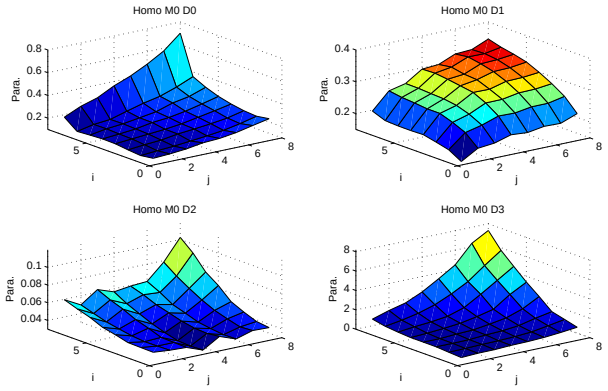



Fig. 5. Parameters $b_k$ ($k = 0, 1, \cdots, 63$) of $8 \times 8$ CB with Homogeneous and M0 classifications.

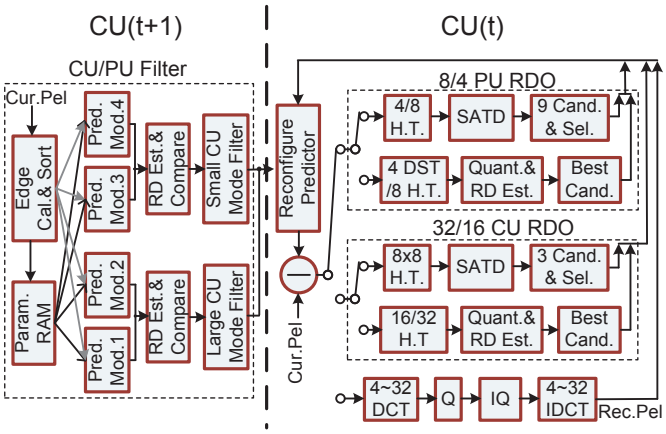



Fig. 6. Top Block Diagram of Proposed HEVC Intra Encoder.

TABLE I
PARAMETER $\omega_r$ VALUES DEFINITION

| $\widetilde{PE}_k/Qs^2$ \ $N$ | 4 | 8 | 16 | 32 |
|---|---|---|---|---|
| [ 0 , 1/8) | 0 | 0 | 0 | 0 |
| [1/8 , 1/4) | 1/8 | 1/2 | 1/8 | |
| [1/4 , 1/2) | 1/4 | 1 | 1/4 | 1/2 |
| [1/2 , 1 ) | 1/2 | 4 | 1/2 | 2 |
| [1 , 2 ) | 1 | 16 | 1 | 8 |
| [2 , 4 ) | | 32 | 2 | 32 |
| [4 , 8 ) | | | 4 | 64 |
| [8 , $\infty$) | | | 16 | 128 |

ing rate $\widetilde{R}_k$ and the distortion $\widetilde{D}_k$ overhead as follows.

$$\begin{cases} \widetilde{R}_k = \dfrac{7\omega_r \cdot \widetilde{PE}_k}{64} \\ \widetilde{D}_k = \omega_d \cdot \widetilde{PE}_k \end{cases} \tag{15}$$

$\omega_r$ and $\omega_d$ are the weighting factors for rate and distortion, respectively. Through theory analysis and experimental feedbacks, the values of $\omega_r$ are defined as showed in Table I, while $\omega_d$ is derived as

$$\omega_d = \begin{cases} 1 & , \quad \widetilde{PE}_k > Qs^2/16 \\ 0 & , \quad otherwies \end{cases} \tag{16}$$

For a $N \times N$ CB, if it is encoded as one partition, its RD-cost ($RD_N$) is estimated as

$$RD_N = \sum_{k=0}^{N^2-1} (\widetilde{R}_k + \widetilde{D}_k). \tag{17}$$

On the other hand, if the CB is partitioned into 4 sub-blocks, the corresponding RD-cost ($RD_{\boxplus N}$) is composed of two terms. The first one is $RD_{N/2}(n)$ for 4 sub-blocks, and second one is the additional side-information (prediction mode and code-block-flag) coding costs. In summary, $RD_{\boxplus N}$ is written as

$$RD_{\boxplus N} = \sum_{n=0}^{3} RD_{N/2}(n) + 3\frac{7}{64}(\gamma_{\text{mode}} + \gamma_{\text{cbf}}), \tag{18}$$

in which, $\gamma_{\text{mode}} = 4$ represents the prediction mode bits, and $\gamma_{\text{cbf}} = 1$ indicates the code-block-flag bit cost. It should be noticed that the $\omega_r$ value in $RD_{N/2}(n)$ is always smaller than that in $RD_N$.

If $RD_N \leq RD_{\boxplus N}$, the current CB is coded as a whole one; Otherwise, we chose sub-partition strategy in the following R-DO search.

## IV. EXPERIMENT RESULTS

The prototype HEVC Intra encoder integrated with our proposals is implemented with TSMC90nm technology. The top block diagram is shown as Fig. 6. The encoder is two CTU pipelined architecture. The first stage eliminates the redundant CU/PU modes by using our proposed methods. The directive information of the first stage is dispatched to the second stage for the RDO processing. The second stage is constituted of 4 prominent components: reconfigurable predictor, $8 \times 8/4 \times 4$ PU RDO engine, $32 \times 32/16 \times 16$ CU RDO engine, and the reconstructed datapath.

Each RDO engine is only assigned one mode, which is generated by the previous stage, to estimate its best RD-cost. It should be noticed that the predictor is shared by two RDO engines. This comes from two factors: First, the predictor is reconfigurable, which produces $L$ modes $N \times 1$ row-wise prediction pixel in each cycle ($N$ is the CB size and $L \times N = 128$); Second, as shown by Fig. 1, during the RD-cost estimation period (marked by the slash), $8 \times 8/4 \times 4$ RDO engine will not occupy the predictor. Therefore, the prediction engine can be used by two RDO engines alternatively. By using the variable block size unified DCT/IDCT architecture, one reconstruction datapath is shared by all CBs, when the best PU mode of a CB is decided.

All the modules for proposed RDO designs are described with Verilog HDL and synthesized with Design Compiler based on TSMC90nm 1P9M technology. The gate count and power dissipation of each primary component are shown in Table II. As compared with primitive parallel implementation (described in Section II), 57% hardware is saved. At the worst conditions (0.9v, 125°C), the maximum clock speed is 357MHz, which fulfills the real-time encoding of HD1080p@44fps. Accordingly, the power dissipation is 217.9mW.

TABLE II
HARDWARE CONSUMPTION OF PROPOSED HEVC INTRA ENCODER

| Module | Pre-Mode Filter | Rcnf. Predictor | 32/16 CU RDO | 8/4 PU RDO | Rcns. Datapath | Total |
|---|---|---|---|---|---|---|
| Gates(K) | 214.1 | 817.3 | 781.3 | 450.6 | 507.2 | 2269.0 |
| Pwr(mW) | 26.2 | 101.4 | 25.2 | 32.9 | 32.2 | 217.9 |

Modules in this talble could be referred to figure 6.

TABLE III
CODING PERFORMANCE COMPARED TO HM-10

| Class | Sequence | BD-PSNR [dB] | BD-Rate [%] | Time Saved [%] |
|---|---|---|---|---|
| A | PeopleOnStreet | -0.21 | 4.61 | 61.4 |
| | Traffic | -0.21 | 4.34 | 61.9 |
| B | BasketballDrive | -0.17 | 6.73 | 61.7 |
| | BQTerrace | -0.19 | 4.32 | 64.9 |
| | Cactus | -0.14 | 4.28 | 72.6 |
| | Kimono | -0.12 | 4.39 | 68.1 |
| | ParkScene | -0.11 | 3.39 | 58.3 |
| | Tennis | -0.18 | 5.92 | 62.0 |
| C | BasketballDrill | -0.21 | 4.63 | 60.0 |
| | BasketballDrillText | -0.21 | 4.75 | 60.6 |
| | BQMall | -0.20 | 4.15 | 58.4 |
| | RaceHorses | -0.19 | 3.38 | 58.6 |
| D | BassketballPass | -0.24 | 4.80 | 58.8 |
| | BlowingBubbles | -0.19 | 3.44 | 54.2 |
| | BQSquare | -0.15 | 1.97 | 55.1 |
| | Keiba | -0.21 | 4.02 | 60.3 |
| E | SlideEditing | -0.41 | 2.94 | 61.1 |
| | Vidyo1 | -0.25 | 6.04 | 64.8 |
| | Vidyo3 | -0.23 | 5.35 | 63.4 |
| | Vidyo4 | -0.21 | 5.19 | 63.6 |
| | Johnny | -0.21 | 5.15 | 63.5 |
| | KristenAndSara | -0.24 | 5.86 | 63.6 |
| Average | | -0.20 | 4.53 | 61.7 |

The coding performance analysis is illustrated by Table III. The original HM reference encoding (low-complexity configuration) is referred as the anchor. Twenty-two typical video sequences were tested with Intra coding configurations and $QP$=$\{$22,27,32,37$\}$. Bjontegaard Delta PSNR/Rate(BD-PSNR/Rate) is adopted to qualitatively measure the coding efficiency of our methods. The computational complexity reduction is measured in terms of the encoding time. As shown in Table III, the proposed methods averagely introduce 0.20dB BD-PSNR coding quality degradation, or equivalently 4.53% bit-rate increasing, while the complexity saving is up to 61.7% on average.

## V. CONCLUSIONS

This paper presents the fast HEVC Intra encoder architecture with the pre-CU/PU mode filtering algorithm friendly to parallel VLSI RDO processing. On average, 61.7% encoding complexity could be saved, while the incurred coding quality loss is 0.20dB BD-PSNR. Moreover, our pre-CU/PU filtering and parallel CU/PU searching design contributes to the realization of real-time encoding system, since it ensures a stable speedup performance without any computation complexity trembling. The encoder is implemented TSMC 90nm CMOS technology. With 357MHz clock speed, the proposed design supports the real-time encoding of 4:2:0 format HD1080p at the frame rate of 44fps.

## REFERENCES


[1] Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan, Ye-Kui Wang, and Thomas Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 10," Geneva, CH, 2013, JCT-VC-L1003.

[2] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[3] J-R Ohm and Gary J Sullivan, "High Efficiency Video Coding: The Next Frontier in Video Compression [Standards in a Nutshell]," *IEEE Signal Processing Mag.*, vol. 30, no. 1, pp. 152–158, 2013.

[4] Mohammed Golam Sarwer and Lai-Man Po, "Fast bit rate estimation for mode decision of h. 264/avc," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1402–1407, 2007.

[5] Yu-Kuang Tu, Jar-Ferr Yang, and Ming-Ting Sun, "Efficient rate-distortion estimation for H.264/AVC coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 5, pp. 600–611, 2006.

[6] Siwei Ma, Shiqi Wang, Shanshe Wang, Liang Zhao, Qin Yu, and Wen Gao, "Low complexity rate distortion optimization for hevc," in *Data Compression Conference (DCC)*. IEEE, 2013, pp. 73–82.

[7] Thaísa L da Silva, Luciano V Agostini, and Luis A da Silva Cruz, "Fast HEVC intra prediction mode decision based on edge direction information," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 1214–1218.

[8] Guilherme Correa, Pedro Assuncao, Luciano Agostini, and Luis A. da Silva Cruz, "Coding Tree Depth Estimation for Complexity Reduction of HEVC," in *Data Compression Conference (DCC)*. IEEE, 2013, pp. 43–52.

[9] Seunghyun Cho and Munchurl Kim, "Fast CU Splitting and Pruning for Suboptimal CU Partitioning in HEVC Intra Coding," *IEEE Trans. Circuits Syst. Video Technol.*, 2013(in press).

[10] Xiaolin Shen, Lu Yu, and Jie Chen, "Fast coding unit size selection for HEVC based on bayesian decision rule," in *Picture Coding Symposium (PCS), 2012*. IEEE, 2012, pp. 453–456.

[11] Su-Wei Teng, Hsueh-Ming Hang, and Yi-Fu Chen, "Fast mode decision algorithm for residual quadtree coding in HEVC," in *Visual Communications and Image Processing (VCIP), 2011 IEEE*. IEEE, 2011, pp. 1–4.

[12] Feng Pan, Xiao Lin, Susanto Rahardja, Keng Pang Lim, ZG Li, Dajun Wu, and Si Wu, "Fast mode decision algorithm for intraprediction in H. 264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 813–822, 2005.