Fast Mode Decision Algorithm for Residual Quadtree Coding in HEVC

Su-Wei Teng[1], Hsueh-Ming Hang[1], and Yi-Fu Chen[2]

[1]Department of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan

[2]Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Taoyuan, Taiwan

***Abstract*—High Efficiency Video Coding (HEVC) is an in-progress next generation video coding standard. It has a similar basic structure to the ITU/MPEG H.264/AVC coder but with further enhancement on each coding tool to increase compression efficiency. Thus, a Residual Quadtree (RQT) coding scheme is adopted in the conventional transform coding procedure. The cost is the additional computation complexity when compared to the fixed-size transform. In this paper, we design a fast algorithm for Residual Quadtree mode decision. Considering the Rate-Distortion efficiency, we replace the original depth-first mode decision process by a Merge-and-Split decision process. Furthermore, because a substantial numbers of zero-blocks are produced after quantization and mode decision, we reduce the unnecessary computation by using the inheritance property of zero-blocks. In addition, for nonzero-blocks, two early termination schemes are developed for both TU Merge and TU Split procedures, respectively. Comparing to HM 2.0, our method saves the RQT encoding time from 42% to 55% for a number of test videos with negligible coding loss.**

I. INTRODUCTION

Recently, ISO-IEC/MPEG and ITU-T/VCEG have formed the Joint Collaborative Team on Video Coding (JCT-VC), which aims to develop the next generation video coding standard, called High Efficiency Video Coding (HEVC). With a flexible coding architecture and respective tool extension from previous standard H.264/AVC, a promising compression performance can be expected.

In this paper, we focus on the transform coding tool in HEVC, the residual quadtree (RQT) coding, in view of computational complexity. The rest of this paper is organized as follows. Section II is a short introduction to RQT including the encoding flow and our research motivation. In Section III, we analyze the characteristics of the nested quadtree-based transform coding and propose a new mode decision architecture. Based on the information in Section III, Section IV presents our fast mode decision algorithm with several early termination schemes. In Section V, the experimental results of our method compared with HM 2.0 [1] are presented. In the end, Section VI briefly summarizes our work

II. OVERVIEW OF RESIDUAL QUADTREE IN HEVC

In HEVC, the nested quadtree-based transform coding has a coding performance superior to the traditional fixed-block-size transform because it can be efficiently adapted to the locally varying characteristic of prediction residuals. Such a technique, so-called *residual quadtree* (RQT) [3], provides a high degree of flexibility by selecting the optimal Transform Unit (TU) partitions with transform sizes ranging from 4x4 to 32x32. Which transform size (partition or *mode*) should be used for a particular image area? Thus, the RQT mode decision plays a key role for the purpose of achieving the optimal performance, as described in [3].

Ideally, the best RQT mode is selected by comparing the Lagrangian Rate-Distortion costs (RD cost) of all possible partitions (modes). Practically, it may be implemented by using a depth-first search procedure, as illustrated by Fig. 1. The search order is labeled using the alphabetical order in Fig. 1; the lower case node label also represents a TU partition mode. Note that the maximum admissible depth is 3 in the Random Access (RA) and High-efficiency (HE) setting of HM 2.0 [2].

Experiments show that the encoding time of a depth-3 RQT is about three times the encoding time of a fixed-size transform coding. Hence, we propose methods to reduce computations on the RQT mode decision procedure in this paper.

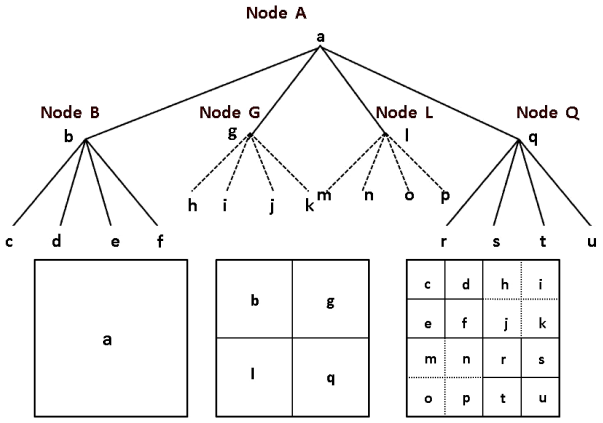

Fig. 1 Diagram of RQT

III. ANALYSES ON NESTED QUADTREE-BASED TRANSFORM CODING

A. Zero-Block Inheritance

In many cases, all DCT coefficients of a TU are quantized to zero in those residual blocks containing little energy. It happens often especially in the still region of a frame. Strong temporal similarity increases the inter-frame prediction accuracy. Given a TU, the RQT mode decision process needs to evaluate two options: 1) encoding residuals, or 2) skipping residual coding. At low bitrates particularly, to save bits, TU skip modes are often chosen after RD optimization. We denote these TUs as "zero-blocks".

In general, a zero-block judged at 32x32 block size does not imply all its 16x16 sub-blocks are zero-blocks. Very often, however, all the subsequent partitions of a zero-block are zero-blocks (of smaller sizes). We call this hypothesis, the Zero-Block Inheritance (ZBI). To be precise, we assume all the quadtree-partitions of a zero-block have strong tendency to be zero blocks, and vice versa.

Base on this hypothesis, the behavior of RD cost of the hierarchical zero-blocks are analyzed. For a zero block, the RD cost of depth-k TU is

$$J_{Zero\ Block,k} = SSE_k + \lambda R_{CBF,k}, \quad (1)$$

where SSE_k is the sum of the pixel squares of residual block at depth k. The bitrate term, R_{CBF}, consists of only the Coded Block Flag (CBF) coding bits (due to zero quantized DCT coefficients), and λ is the Lagrangian multiplier. Meanwhile, based on ZBI, its quadtree-partitions are also all zero blocks. Hence, the RD cost sum of the four sub-blocks is

$$\sum_{i=1}^{4} J_{Zero\ Block,k+1,i} = \sum_{i=1}^{4} SSE_{k+1,i} + \lambda \sum_{i=1}^{4} R_{CBF,k+1,i}. \quad (2)$$

Because both depth-k TU and depth-(k+1) subTUs share the same residual block, $SSE_k = \sum_i^4 SSE_{k+1,i}$. In (1) and (2), λ is typically a constant value decided by Qp. In general, the coding bits of CBF in (1) are lower than the sum of its quadtree-partitions CBF bits in (2). Thus,

$$J_{Zero\ Block,k} \le \sum_{i=1}^{4} J_{Zero\ Block,k+1,i}. \quad (3)$$

Similarly, the relation in (3) can be extended to depth k+2 or higher. Based on the above arguments, we draw a conclusion that once a TU is found to be a zero-block, we terminate the tree partition process and choose the zero-block mode at this depth.

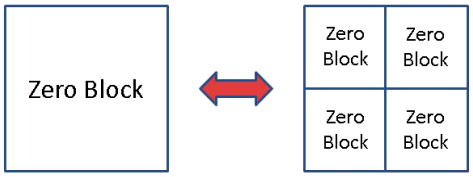

Fig. 2 Diagram of Zero-Block Inheritance

B. Statistics of Transform Unit Partition

To verify our Zero-Block Inheritance conjecture in the last sub-section, we collect the statistics of TU partitions generated by HM 2.0. Given the current block-size, Table 1 shows the (conditional) probabilities of the final decision that the block is non-split (no further partition). In other words, in some rare cases, even though the current block is zero-block, the optimal modes of its sub-partitions are not all zero-blocks and thus it may be further split. In general, the conditional probabilities are higher than 0.82 on the average. Especially, the 8x8 and 16x16 TUs have even higher values. In other words, a 90% or higher confidence is achieved if we terminate the mode search calculations when a zero-block is detected.

On the other hand, given that the current block is a nonzero-block, the probabilities that the optimal mode is non-split are tabulated in Table 2. In other words, they show the probabilities of no further partitions. Obviously, the probabilities are much smaller than those in Table 1. Furthermore, statistic shows that the nonzero-block conditional probabilities of 32x32 TU are much lower than those of small bock sizes. That is, the 32x32 TU is selected much less compared to the 8x8 and 16x16 TUs given that that TU is a nonzero block.

Table 1. Zero-block TU partition mode distribution

Qp22	Sequence					
TU Size	S09	S10	S12	S14	S15	Avg.
32x32	0.88	0.78	0.93	0.75	0.74	0.82
16x16	0.93	0.91	0.96	0.89	0.87	0.91
8x8	0.94	0.90	0.97	0.90	0.90	0.91
Qp32	Sequence					
TU Size	S09	S10	S12	S14	S15	Avg.
32x32	0.95	0.93	0.95	0.92	0.82	0.92
16x16	0.98	0.97	0.98	0.97	0.93	0.97
8x8	0.99	0.97	0.99	0.98	0.97	0.98

Table 2. Nonzero-block TU partition distribution

Qp22	Sequence					
TU Size	S09	S10	S12	S14	S15	Avg.
32x32	0.22	0.16	0.14	0.10	0.16	0.16
16x16	0.53	0.44	0.44	0.42	0.41	0.45
8x8	0.61	0.51	0.57	0.50	0.58	0.55
Qp32	Sequence					
TU Size	S09	S10	S12	S14	S15	Avg.
32x32	0.34	0.25	0.22	0.30	0.24	0.27
16x16	0.63	0.54	0.54	0.58	0.52	0.56
8x8	0.74	0.61	0.71	0.70	0.70	0.69

Test sequence:S09/BQMall; S10/PartyScene; S12/BasketballPass; S14/BlowingBubbles; S15/RaceHorses

C. Priority of TU Size in Mode Decision

The HM 2.0 depth-first RQT decision process adopts the top-down search order. It always starts from the maximum admissible size TU in a CU leaf and evaluates the possibility of further partitions. But from the earlier discussions, we conclude that the 32×32 TU does not offer as much rate-distortion efficiency as the smaller TUs. Thus, we force the 32x32 TU to be the last candidate in the mode search procedure if it appears. By doing this, we are able to save computing power by skipping unnecessary TU partition evaluation. The modified RQT mode decision algorithm becomes a TU split-and-merge process as illustrated in Fig. 3

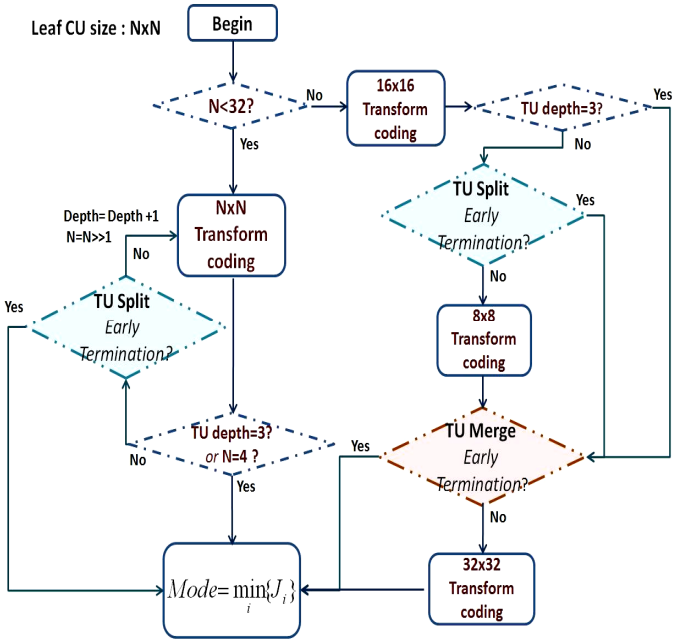

Fig. 3 TU Split-and-Merge RQT mode decision algorithm

IV. Fast RQT Mode Decision Algorithm

In this section, we propose a fast RQT mode decision algorithm depicted in Fig. 3. Depending on the size of leaf CU, there are two paths can be chosen. As the leaf CU size is smaller than 32x32, the left path in Fig. 3 is chosen, which recursively evaluates whether to further partition TU into subTUs. This procedure is denoted as "TU Split" decision process. On the other hand, when the leaf CU size is 64x64 or 32x32, the 16x16 TU partition is first evaluated as showed by the right path. This path evaluates the 32x32 TU mode in the last. This final step is denoted as "TU Merge" since this process determines whether the smaller subTUs should be merged into a larger 32x32 TU or not. Furthermore, two early termination schemes are proposed for both TU Merge and TU Split steps, respectively. These schemes are described below.

A. Early Termination Scheme for TU Merge Process

(1) ZBI Early Termination for TU Merge

If all the four 16x16 subTUs are zero-blocks, the evaluation of TU Merge is not executed.

(2) Block Adaptive RD Cost Threshold

For the purpose of skipping unnecessary TU Merge process, an adaptive RD cost threshold is calculated using the information of four 16x16 subTUs.

(a) **Bitrate Prediction**: According to the collected data, a strong correlation exists between the Merge TU bitrate and the bitrate sum of its four subTUs. Thus, the bitrate of Merge TU can be estimated using the following linear regression formula.

$$\hat{R}_{Merge\ TU} = \beta(\sum_{i=1}^{4} R_{subTU,i}) + \alpha, \tag{4}$$

where the two repression parameter β is 0.9 and α has a larger variation and is nearly zero as derived from the training sequences (Table 3).

Table 3. Empirical linear regression parameter

	S09	S10	S14	S15	AVG.
α	4.2	-8.2	14.6	1.0	2.9
β	0.90	0.94	0.86	0.93	0.91

S09/BQMall; S10/PartyScene; S14/BlowingBubbles; S15/RaceHorses

(b) **Block-Level Adaptive Distortion Model**: To construct an analytic distortion model, we assume the prediction residual signals are the (block-wise) WSS first-order Markov random field with zero-mean. Thus, the spatial domain statistics can be estimated at block-level adaptively.

Based on the previous work in [4] and [5], we adopt a linear relation between the spatial domain and the transform domain representations as follows.

$$\sigma^2_{F,uv} = \sigma^2_f [CRC^T]_{u,u} [CRC^T]_{v,v}, R = \begin{bmatrix} 1 & \rho & \cdots & \rho^{N-1} \\ \rho & 1 & \cdots & \rho^{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \cdots & 1 \end{bmatrix}_{NxN}, \tag{5}$$

where $\sigma^2_{F.uv}$ is the variance of the $(u,v)^{th}$ DCT coefficient, C is the N by N DCT matrix, σ^2_f is the block variance, and ρ is the correlation coefficient between neighboring pixels. As a result, the DCT coefficients in the same block have zero-mean Gaussian distributions [6] with variance $\sigma^2_{F.uv}$ defined by (5).

With the above model, the quantization error of the $(u,v)^{th}$ DCT coefficient can be calculated using the quantizer model. Furthermore, the total distortion can be estimated by a function in (6).Given the block size N and quantization parameter Qp, this model estimates the block distortion individually by using two spatial domain parameters, block variance σ^2_f and correlation coefficient ρ,

$$D = \sum_{u,v} D_{u,v} = f(\sigma^2_f, \rho, N, Qp) = f(\sigma^2_f, \rho), \forall Qp, N. \tag{6}$$

Fig. 4 shows the estimated distortion using (6) for 16x16 TU, Qp =22 and 32, and for the correlation coefficient ranging from 0.3 to 0.6. However, the curves in Fig. 4 cannot be represented by simple mathematical functions. For practical usage, we simplify the original distortion model by using piece-wise line segments as shown in Fig. 5. Since the correlation coefficient controls the curvature of the distortion model, in the simplified model, the slope K in the linear segment is roughly inversely proportional to ρ and is estimated by the distortion of subTUs. The estimator of K value is

$$\hat{K} = \frac{D_{subTU}}{\min(\sigma^2_f, p_{saturated})}, \tag{7}$$

where D_{subTU} is the distortion of a subTU obtained before TU Merge process and $p_{saturated}$ is the saturation point in our simplified model, which depends on Qp. To use the information of the subTUs, the minimum K values among the four subTUs is picked as a lower bound for the K value of the merged TU. Substituting the lower bound into the simplified distortion model, the threshold for TU Merge is calculated as follows.

$$D_{TH} = \min\{\hat{K}_i\} \times \min\{\hat{\sigma}^2_{f,Merge\ TU}, p_{saturated}\}, i = 1 \sim 4, \tag{8}$$

where $\hat{K}_i$ is the K value of the i^{th} subTU, and $\hat{\sigma}^2_{f,Merge\ TU}$ is the sample block variance of the merged TU.

(c) **RD Cost Threshold**: Using the estimated bitrates and the distortion threshold derived from the information of subTUs, the block adaptive RD cost threshold J_{TH} becomes

$$J_{TH} = D_{TH} + \lambda \hat{R}_{Merge\ TU}. \tag{9}$$

When the RD cost of the current best partition mode is lower than J_{TH}, the TU Merge mode decision process is skipped.

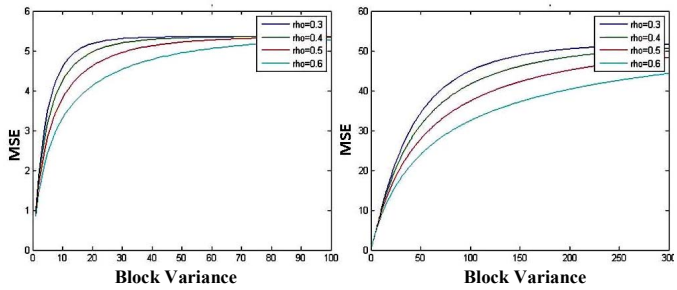

Fig. 4 Block distortion model for *Qp22(left)and Qp32(right)*

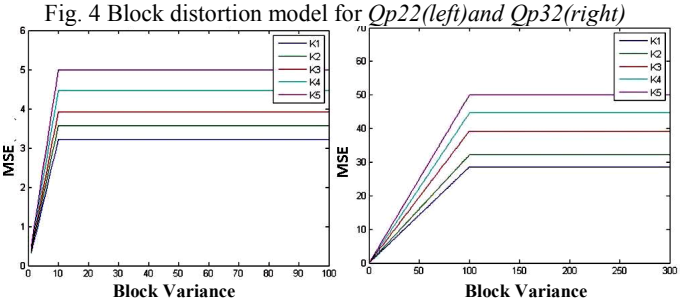

Fig. 5 Simplified block distortion model for *Qp22(left)and Qp32(right)*

B. Early Termination for TU Split Process

(1) ZBI Early Termination for TU Split

If the current TU is found to be a zero-block, no more evaluation on further TU partition is executed.

(2) RD Cost Prediction by Partial SubTUs

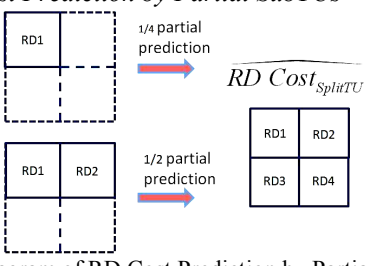

Fig. 6 Diagram of RD Cost Prediction by Partial SubTUs

To be able to terminate the TU Split process early, we propose a strategy, which dynamically predicts the RD cost of TU Split by using the information of partial subTUs while the TU Split process is underway. To this end, we define a *proportion factor*, denoted as p_k,

which defines the ratio between the TU Split RD cost and the RD cost of the partial subTUs. The definition of p_k, is given below, which is the percentage of the accumulated RD cost up to the k^{th} subTUs.

$$p_k = \frac{\sum_{i=1}^{k} RD\ Cost_{i,subTU}}{\sum_{j=1}^{4} RD\ Cost_{j,subTU}}, \text{k=1,2,3.} \quad (10)$$

Obviously, the value of p_k varies with TU. Hence, a statistical model is needed to describe the randomness of p_k. Our experimental data show that the distribution of p_k is somewhere between the Gaussian distribution and the Laplacian distribution as shown in Fig. 7. Two distributions p_1 and p_2, representing the proportion factors of one subTU and two subTUs, respectively, are illustrated by collecting data from the sequence "PartyScene" with *Qp 22*. Because only the first and the second moments are needed to determine all the parameters in the Gaussian distribution and the Laplacian distributions, we calculate the mean and the variance of p_1, p_2 and p_3 from the training sequences and the results are listed in Table 4.

Have obtained the distribution parameters of p_k, we pick up the RD cost threshold for TU Split using (11)

$$J_{TH,Split\ TU} = \frac{\sum_{i=1}^{k} RD\ Cost_{i,subTU}}{\widehat{p_k}}, k = 1,2,3, \quad (11)$$

where $J_{TH,Split\ TU}$ is the TU Split RD cost threshold. In order not to overestimate the RD cost of TU Split, $\widehat{p_k}$ is estimated with mean $\widehat{m}_{p_k}$ plus one or two standard deviation $\widehat{\sigma_{P_k}}$; that is

$$\widehat{p_k} = \widehat{m}_{p_k} + n\widehat{\sigma_{P_k}}, \quad (12)$$

where n is 2 for k=1, and n=1 as k=2,3. By doing this, an 80% or higher confidence for estimating the RD cost for TU Split is achieved. When the RD cost of the current best partition mode is lower than $J_{TH,Split\ TU}$, the TU Split process is terminated.

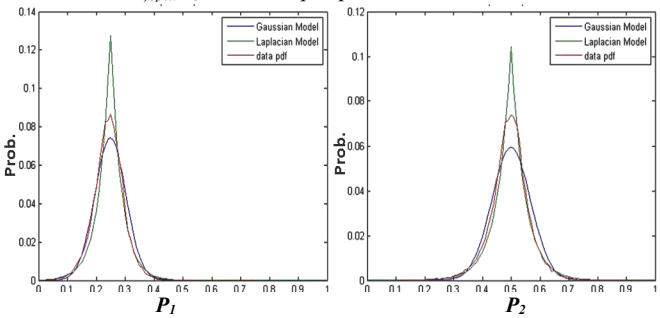

Fig. 7 The distribution of P_1 and P_2 compared with the Gaussian and Laplacian distribution in the same parameters

Table 4. Mean and variance of p_1, p_2 and p_3

	Mean $\widehat{m}_{p_k}$	Variance $\widehat{\sigma}^2_{P_k}$
p_1	0.25	0.011
p_2	0.50	0.015
p_3	0.75	0.011

V. Experimental Results

In the following experiments, our proposed algorithm is implemented on the HM 2.0 software [1] with the Random Access and High-efficiency configuration [2]. And four *Qp* values are tested, which are 22, 27, 32 and 37(denoted as Qp_1, Qp_2, Qp_3 and Qp_4). The test video sequences cover a broad range of visual characteristics with two different resolutions.

As summarized in Table 5, our proposed schemes are compared with HM 2.0 [1] in terms of BDP (dB), BDR (%), TS (%).The first two measurements, BDP and BDR, are the Bjontegaard metrics [7], which present the average relative differences in RD performance. The TS is the RQT time saving using the proposed method and is defined as

$$TS = \frac{1}{4}\sum_{i=1}^{4} \frac{RQT\ time_{HM2.0}(Qp_i) - RQT\ time_{proposed}(Qp_i)}{RQT\ time_{HM2.0}(Qp_i)} \times 100\%,$$

where $RQT\ time_{HM2.0}(Qp_i)$ is the RQT encoding time of HM 2.0 and $RQT\ time_{proposed}(Qp_i)$ is the encoding time of our scheme built on HM 2.0 with $Qp = Qp_i$.

The simulation data show that our scheme can provide averagely 49% overall time saving (TS) over HM 2.0 RQT, which is about 2 times faster in the encoding process. On the average, the BD-rate increase is about 0.31% and the BD-PSNR Y loss is about 0.013 dB. The impact of using the proposed fast algorithm on the RD performance is negligible.

Table 5. Performance Comparison

Sequence		BDR	BDP	TS
Class C	BasketballDrill	0.36	-0.014	52
	BQMall	0.27	-0.009	55
	PartyScene	0.18	-0.008	48
	RaceHorses	0.42	-0.016	42
Class D	BasketballPass	0.35	-0.017	49
	BQSquare	0.31	-0.013	55
	BlowingBubbles	0.31	-0.013	51
	RaceHorses	0.26	-0.012	42
AVG.		**0.308**	**-0.013**	**49.25**

VI. Conclusions

In this paper, we propose a fast algorithm that eliminates the unnecessary transform coding examinations for RQT mode decision in HM 2.0 [1]. The proposed method first checks the partition mode that provides higher rate-distortion efficiency. Based on the information collected in executing the checks, we construct three early termination thresholds. Some thresholds are data dependent for individual blocks. In general, our method achieves up to 55% encoding time reduction with negligible coding loss, as compared to HM 2.0 [1].

VII. Acknowledgment

This work was supported in part by the Chung-Hwa Telecomm, Taiwan under Grants TL-100-G113 and by the NSC, Taiwan under Grants 98-2221-E-009 -076.

References

[1] JCT-VC, "High Efficiency Video Coding (HEVC) Test Model 2 (HM 2) Encoder Description", JCTVC-D502, April 2011.

[2] JCT-VC, "Common test conditions and software reference configurations", JCTVC-D600, February 2011.

[3] D. Marpe *et al.* "Video Compression Using Nested Quadtree Structures, Leaf Merging, and Improved Techniquesfor Motion Representation and Entropy Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1676–1687, 2010.

[4] I. M. Pao and M. T. Sun, "Modeling DCT coefficients for fast video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 608–616, 1999.

[5] Z. Fan, O. C. Au, and N. M. Cheung, "Transfrom-Domain Adaptive Correlation Estimation (Trace) for Wyner-Ziv Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1423–1436, 2010.

[6] E. Y. Lam and J.W. Goodman, " A Mathematical Analysis of the DCT coefficient Distribution for Images," *IEEE Trans. Image Procss.*, vol.9 , no. 10, pp1661-1666, 2000

[7] G. Bjontegaard, "Calculation of Average PSNR Differences between RD-curves," Document VCEG-M33, Apr. 2001.