A CONTEXT-BASED ADAPTIVE FAST INTRA_4x4 PREDICTION MODE DECISION ALGORITHM FOR H.264/AVC VIDEO CODING

Kai Zhang[1, 2], *Qiang Wang*[1], *Qian Huang*[1, 2], *Debin Zhao*[1], *Wen Gao*[2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
[2] Graduate University of the Chinese Academy of Sciences, Beijing 100039, China
{kzhang, qwang, qhuang, dbzhao, wgao}@jdl.ac.cn

ABSTRACT

In this paper, we propose a context-based adaptive fast Intra_4x4 prediction mode decision algorithm for H.264/AVC. Firstly, a 3-order Markov random field model is introduced to describe the spatial distribution of RD optimal Intra_4x4 prediction modes at picture level. Secondly, in a neighboring context specified by this model, only a small set of candidate modes are chosen for RDO calculation according to a constrained error probability criterion. Thirdly, to reduce candidate modes further, thresholds which can be estimated by linear regression are adopted to perform early terminations. Finally, the proposed algorithm can adapt itself well to diverse video sequences due to an ability of automatic adjustment, while the model and thresholds are initialized off-line by statistics. No apparent computational overheads are involved throughout the algorithm. Experiments show that the novel fast algorithm searches only 2 - 6 Intra_4x4 modes per block and reduces up to 55% encoding time with a penalty on coding efficiency less than 0.1 dB.

Index Terms - H.264, intra-prediction, fast algorithm

1. INTRODUCTION

H.264/AVC is an excellent and state-of-the-art international video coding standard [1]. Compared with previous standards, H.264/AVC improves the coding performance significantly at the expense of a much higher computational complexity [2]. In addition, the Lagrangian Rate-Distortion Optimization (RDO) [3] method increases the computational burden extremely at the encoder side. To reduce the complexity, fast intra-prediction decision algorithms have drawn numerous academic attentions.

Several fast intra-prediction decision algorithms have been studied. Pan *et al.* [4] proposed an algorithm which utilized the local edge direction information to select a small number of candidate intra-prediction modes. Meng *et al.* [5] believed that Intra_4x4 prediction modes with similar directions to the most probable mode should be the candidate ones. Kim *et al.* [6] filtered intra-prediction modes with two joint features, *i.e.*, SAD and SATD values and proposed a fast RD cost estimation algorithm.

Although previous researches on fast intra-prediction decision have shown pretty good results, we can still make improvements in several aspects. Firstly, most algorithms bring serious computational overheads such as calculations of edge directions, SADs and SATDs. Secondly, it is still a problem to establish reasonable and effective thresholds, which are widely used for early terminations. Finally, fast algorithms suffer an apparent performance loss compared with the exhaustive one. In all I-frames coding, a loss about 0.3 dB is reported [4] [6].

In this paper, a novel context-based adaptive fast Intra_4x4 prediction decision algorithm is presented. Intuitively, since neighboring blocks have a strong correlation, we can estimate the Intra_4x4 prediction mode of the current block based on modes of coded neighboring blocks. To be more operational, we model the spatial distribution of RD optimal Intra_4x4 prediction modes in a picture as a Markov random field. With conditional probabilities learned off-line, the proposed algorithm skips several modes conforming to a constrained error probability criterion. Besides, thresholds are also adopted for early terminations. With linear regression analysis, we find out an exponential relationship between QPs and thresholds. In addition, the algorithm can adapt itself to a variety of sequences. In an updating process, conditional probabilities are refined, and thresholds are adjusted according to a reward and punishment strategy. Throughout the algorithm, there are no serious computational overheads. Experiments show that the proposed algorithm can reduce computational complexity significantly with a negligible loss in coding performance.

The remainder of this paper is organized as follows. Section 2 reviews the Intra_4x4 prediction in H.264/AVC. Section 3 describes the proposed fast Intra_4x4 prediction decision algorithm. Experimental results are presented and discussed in section 4. Section 5 concludes the paper.

2. INTRA_4x4 PREDICTION IN H.264/AVC

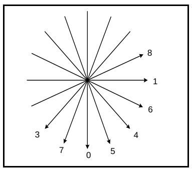


Fig.1. Intra_4x4 prediction mode directions.

Intra_4x4 and Intra_16x16 are two types of luma intra-prediction in H.264 baseline profile and main profile [1]. Intra_4x4 has up to nine prediction modes, namely mode 0 - mode 8. Except the DC mode

(mode 2), the other eight modes correspond to different spatial prediction directions as illustrated in Fig. 1.

In the current H.264/AVC reference software [7], the encoder encodes a 4x4 block with all the nine Intra_4x4 prediction modes exhaustively to determine the best one. For mode m from 0 to 8, the RD cost is calculated as

$$RDCost_{4\times4}(m) = SSD_{4\times4}(m) + \lambda_{mode} \bullet Rate_{4\times4}(m)\,, \qquad (1)$$

where $SSD_{4x4}(m)$ and $Rate_{4x4}(m)$ are the sum of squared distortion and the number of coding bits in this 4x4 block under mode m respectively. The mode with the minimum RD cost is chosen to be the best one, which is defined as the 'accurate' mode in this paper.

3. PROPOSED ALGORITHM

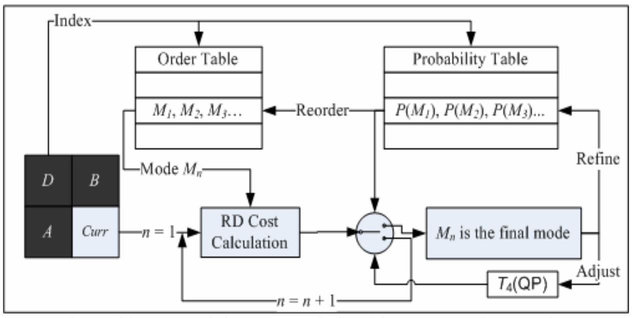
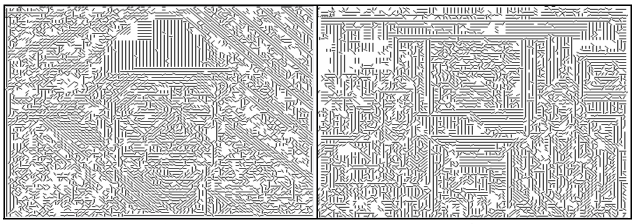


Fig.2. The architecture of the proposed algorithm. '*Curr*' denotes the current 4x4 block. *A*, *B* and *D* are coded neighboring blocks.

We design the fast Intra_4x4 prediction mode decision algorithm in four steps. Firstly, based on a Markov random field hypothesis, we do statistics on several video sequences to obtain mode probabilities in different neighboring contexts, and sort the modes in probability descending order. The probabilities and the order of modes are stored in Probability Table and Order Table respectively, which are both indexed by neighboring contexts. Secondly, when encoding a 4x4 block in a specific neighboring context, the RD cost of each mode is checked one by one in the order indicated by Order Table, until the probability that the accurate mode has not been checked is low enough. Thirdly, after checking each mode, an early termination will happen if its RD cost is lower than a threshold. All the following modes will be skipped in this case. With linear regression, thresholds can be estimated by QPs. Finally, after the mode with the minimum RD cost is chosen to be the final one, an updating process refines the two tables and adjusts thresholds according to a reward and punishment strategy. Fig.2 depicts the architecture of the proposed algorithm.



(a) (b)

Fig.3. Accurate Intra_4x4 prediction modes of all 4x4 blocks on Frame 0 of sequences 'Foreman' (a) and 'News' (b). We use directions of lines to represent Intra_4x4 prediction modes as depicted in Fig.1. And the blank block stands for the DC mode.

3.1. Markov random field

In natural sequences, neighboring blocks usually hold similar geometry properties. Since the accurate Intra_4x4 prediction mode represents the edge direction of this block [4], we deduce that accurate Intra_4x4 prediction modes of adjacent blocks possess a strong correlation. Fig.3 shows the spatial distribution of accurate Intra_4x4 prediction modes at high bit-rates in two sequences.

To characterize the correlation, we model the distribution of accurate Intra_4x4 prediction modes in a picture as a 3-order Markov random field. In this model, the probability that a mode is the accurate one of the current block depends on the accurate modes of coded neighboring blocks. In a formulation way, we define

$$\begin{aligned} &P(M_n \,|\, (M_A, M_B, M_D)) \equiv \\ &P(Mode_{Curr} = M_n \,|\, Mode_A = M_A, Mode_B = M_B, Mode_D = M_D), \end{aligned} \qquad (2)$$

where $Mode_{Curr}$, $Mode_A$, $Mode_B$ and $Mode_D$ are random variables that represent the accurate Intra_4x4 prediction modes of the current block and three neighboring blocks as depicted in Fig.2. M_n, M_A, M_B and M_D are their possible values respectively. It is by definition that $M_n \in \{0, 1, \ldots, 8\}$. To tackle the boundary case, we let M_A, M_B, $M_D \in \{-1, 0, \ldots, 8\}$, where -1 denotes the 'inexistence'. For example, $M_A = -1$ means that block A does not exist because the current block is at the left boundary in a picture. We define the triple (M_A, M_B, M_D) as the 'neighboring context'. At the encoder side, Intra_4x4 type is always checked no matter what the final type of a macro-block is, so neighboring contexts are accessible.

Based on laws of large numbers, we can estimate conditional probabilities in different neighboring contexts from statistical frequencies. Eight sequences in CIF format, *i.e.*, 'City', 'Crew', 'Foreman', 'Harbour', 'News', 'Mobile', 'Stefan' and 'Tempete' are included in the statistics. Table 1 reveals some statistical results in context (0, 0, 0). It can be seen that probabilities of different modes are obviously biased. After estimating conditional probabilities with average frequencies, we sort modes in probability descending order. These orders denoted by $\{M_n(M_A, M_B, M_D)\}$ are stored into Order Table indexed by neighboring contexts. For example, the item (0, 0, 0) records {0, 5, 7, 1, 2, 4, 6, 3, 8} as implicated in Table 1. In addition, the corresponding probabilities are stored into Probability Table. Instead of precise values, integers $\{U_n(M_A, M_B, M_D)\}$ are recorded as

$$U_n(M_A, M_B, M_D) = \lfloor S \cdot P(M_n \,|\, (M_A, M_B, M_D)) \rfloor\,, \qquad (3)$$

where S is a constant number defined as the 'stable factor'. Empirically, we set S to be 30% of the total number of the 4x4 blocks to be encoded.

Table 1. The percentage of each Intra_4x4 mode in context (0, 0, 0).

	0	1	2	3	4	5	6	7	8
City	79.43	3.71	2.86	1.87	2.39	3.32	2.3	2.96	1.15
Foreman	74.87	3.99	2.71	2.62	3.22	3.70	3.92	3.39	1.57
Harbour	80.76	2.34	2.84	1.81	1.65	4.02	1.20	4.13	1.23
Average	80.44	2.95	2.50	1.85	2.11	3.49	2.03	3.47	1.16

3.2. Constrained error probability

In a practical fast Intra_4x4 prediction mode decision algorithm, we cannot avoid error cases in which the accurate mode happens to be skipped. Given a fixed error probability, the proposed algorithm can minimize the number of modes to be checked with a greedy algorithm. In a specific neighboring context (M_A, M_B, M_D), the algorithm checks modes in order $\{M_n(M_A, M_B, M_D)\}$ indicated by Order Table and stops if

$$\sum_{i=1}^{n} U_i(M_A, M_B, M_D) \geq \gamma \cdot \sum_{i=n+1}^{9} U_i(M_A, M_B, M_D)\,, \tag{4}$$

where n is the number of modes checked already, and γ is a positive constant integer defined as the 'tense factor'. Float division computations are avoided by (4), and the error probability P_{err} can be calculated as

$$P_{err} \approx \frac{1}{\gamma + 1}\,. \tag{5}$$

By tuning γ, the proposed algorithm can be complexity scalable. If γ decreases, the computational complexity will decline while the loss in coding performance will deteriorate.

Due to errors, the neighboring context (M_A, M_B, M_D) acquired in practice may be only an approximation. Since P_{err} is usually small enough, this approximation is reliable in most cases.

3.3. Thresholds determination

We utilize thresholds to reduce candidate Intra_4x4 prediction modes further. On the one hand, if the RD cost of the Intra_16x16 type is lower than a threshold T_{16}, the Intra_4x4 type will be abandoned in this macro-block, and all Intra_4x4 prediction modes will be skipped. On the other hand, if the RD cost of Intra_4x4 prediction mode M_n is lower than a threshold T_4, M_n will be the final mode for this block and all M_ks with $k > n$ will be skipped. Since Intra_4x4 type can be abandoned in some macro-blocks, (M_A, M_B, M_D) may be inaccessible in some blocks. In these blocks, the traditional exhaustive mode decision algorithm will be used.

Relationships between QPs and reasonable thresholds are explored with the linear regression analysis. We take T_4 as an example. At first, we say that a 'shoot' event happens when T_4 takes effect; and a 'hit' event happens when the mode chosen by T_4 is just the accurate one. Then we define α as the 'hit ratio', which represents the ratio of the number of times a hit event happens to the number of times a shoot event happens. Obviously, α is a decreasing function of T_4, and vice versa. Statistically, we are highly confident that a hit event will happen if the hit ratio is above 0.95. Thus the T_4 with α = 0.95 is considered as a reasonable threshold, denoted as T_4'. To explore the relationship between T_4's and QPs, we test several sequences to record T_4's at different QPs. Fig.4 (a) shows the result of the sequence 'Tempete' in CIF format as an example. Apparently, an approximate exponential relationship exists. Thus we suppose

$$Y = a_4 \cdot X + b_4\,, \tag{6}$$

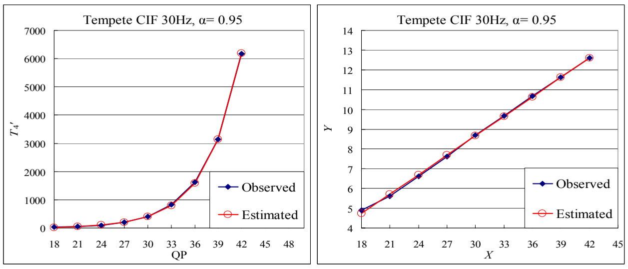
where $Y = log_2 T_4^{Init}$, X = QP and T_4^{Init} is an estimate of T_4'. From (6), we can determine

$$T_4^{Init} = 2^{a_4 \cdot \mathrm{QP} + b_4}\,. \tag{7}$$

By linear regression analysis on the 'Tempete', 'Foreman' and 'News' sequences all in CIF format, we find a_4 = 0.330 and b_4 = -1.265. Fig.4 exposes that the estimated data and the observed data can fit nicely. In the same way, we also determine T_{16}^{Init} as

$$T_{16}^{Init} = 2^{a_{16} \cdot \mathrm{QP} + b_{16}}\,, \tag{8}$$

where a_{16} = 0.311 and b_{16} = 2.981. (7) and (8) are calculated only once at the beginning of the encoding process to obtain the initial values of thresholds.



(a) (b)

Fig.4. (a) The relationship between QPs and T_4's (b) The relationship between X and Y.

3.4. Automatic adjustments

Since different sequences possess different properties, automatic adjustments are necessary. After encoding every T_u 4x4 blocks, an updating process is employed. T_u is defined as the 'updating period'. The updating system may not track the variance in a sequence synchronously if the updating period is too long. While the computational overhead will be unaffordable if the updating period is too short. After taking extensive experiments, we choose T_u to be 50 which shows a good balance between the two sides. In an updating process, termination criteria in 3.2 and 3.3 are invalidated. All Intra_4x4 modes are checked in the order indicated by Order Table to obtain the accurate mode M_k. Then the tables and thresholds are updated.

On the one hand, Probability Table and Order Table are refined and reordered. $U_k(M_A, M_B, M_D)$ is updated as

$$U_k^{New}(M_A, M_B, M_D) = U_k^{Old}(M_A, M_B, M_D) + \delta\,, \tag{9}$$

where δ is defined as the 'convergence factor', and is chosen to be 5. If

$$U_k^{New}(M_A, M_B, M_D) > U_{k-1}^{Old}(M_A, M_B, M_D) \text{ and } k > 1, \tag{10}$$

the values of $U_{k\text{-}1}(M_A, M_B, M_D)$ and $U_k(M_A, M_B, M_D)$ will be switched. The values of $M_{k\text{-}1}(M_A, M_B, M_D)$ and $M_k(M_A, M_B, M_D)$ in Order Table will also be exchanged.

On the other hand, thresholds are adjusted according to a reward and punishment strategy. We still take T_4 as an example. If RD costs of all Intra_4x4 prediction modes are higher than T_4, we will adjust T_4 as

$$T_4^{New} = T_4^{Old} + 2 \cdot \varepsilon\,, \tag{11}$$

where ε is defined as the 'expand factor'. Otherwise, if the RD costs of one or more modes are lower than T_4, and M_h is the first one of them, we will adjust T_4 as

$$T_4^{New} = \begin{cases} T_4^{Old} + \varepsilon & \text{if } k = h \\ T_4^{Old} \cdot \xi & \text{if } k \neq h \end{cases}, \tag{12}$$

where ξ is defined as the 'contract factor'. Empirically, we choose $\varepsilon = 0.03 T_4^{init}$, and $\xi = 0.4$.

4. EXPERIMENTAL RESULTS

The proposed algorithm is implemented on JM12.2 [7] and the original JM encoder is treated as an anchor. We carry out experiments in the H.264 baseline profile for both all I-frames sequences and IPPP sequences without CABAC or 8x8 transforms. At each QP, we record the average number of candidate Intra_4x4 prediction modes per 4x4 block (‘*mpb*’) and the change of encoding time ($\Delta Time$) [4]. $\Delta Time$ is calculated as

$$\Delta Time = \frac{Time_{proposed} - Time_{anchor}}{Time_{anchor}} \times 100\% , \qquad (13)$$

where $Time_{anchor}$ and $Time_{proposed}$ are the total encoding time of the anchor and the proposed encoder respectively. In addition, the average differences in PSNRs (ΔPSNR) and in bit-rates (ΔBits) are calculated [8]. Whole sequences are tested in CIF or 4CIF format.

Table 2 shows results for all I-frames sequences with varying γ. It can be seen that the proposed algorithm checks only 1.66 - 5.95 modes in the 9 Intra_4x4 prediction modes per 4x4 block on average. We can save up to 55.5% encoding time with a penalty in coding efficiency less than 0.1 dB. Although the training data set contains only several sequences in CIF format, the algorithm still shows satisfying results on sequences in 4CIF format or out of the training set. In addition, since less details of texture are retained, neighboring blocks possess a higher similarity at lower bit-rates. Therefore, the proposed algorithm is more efficient as QP increases. Fig.5 (a) and (b) depict two examples of RD performance for all I-frames sequences.

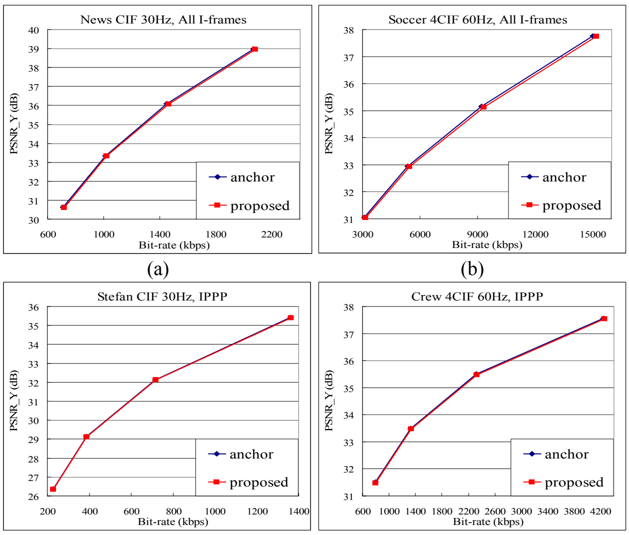
Table 2. The results for all I-frames sequences.

Seq. \ QP	γ	*mpb*				Δ*Time* (%)				ΔPSNR (dB)	ΔBits (%)
		28	32	36	40	28	32	36	40		
Soccer_4cif	10	4.62	3.62	2.48	1.66	-25.6	-34.5	-46.3	-55.5	-0.097	2.29
Container_cif	12	4.38	3.43	2.62	2.12	-27.7	-35.6	-43.7	-50.2	-0.087	1.45
Bus_cif	16	5.95	5.40	4.60	3.66	-13.0	-17.0	-24.2	-32.9	-0.096	1.39
Football_cif	18	5.82	4.99	3.92	2.90	-14.2	-21.1	-31.0	-41.3	-0.090	1.66
Tempete_cif	18	5.95	5.59	5.09	4.42	-11.2	-14.2	-19.0	-25.6	-0.094	1.27
City_4cif	18	5.80	5.02	4.08	3.14	-14.8	-21.3	-30.0	-39.3	-0.084	1.44
Crew_4cif	18	4.32	3.30	2.45	1.75	-27.0	-37.1	-46.4	-54.7	-0.092	2.10
Soccer_cif	18	5.52	4.32	2.94	2.05	-16.4	-26.9	-40.4	-50.6	-0.080	1.93
Harbour_4cif	20	5.73	5.25	4.55	3.69	-17.8	-21.0	-27.1	-34.8	-0.098	1.42
Stefan_cif	20	5.27	5.02	4.68	4.30	-15.4	-17.6	-21.5	-25.8	-0.096	1.14
Foreman_cif	24	5.39	4.54	3.76	3.02	-16.9	-23.9	-32.1	-40.5	-0.096	1.73
News_cif	40	4.34	3.74	3.29	2.84	-21.9	-29.5	-34.9	-40.4	-0.091	1.16
Carphone_cif	50	4.64	3.94	3.26	2.69	-24.3	-30.4	-38.1	-44.5	-0.092	1.38

Table 3. The results for IPPP sequences, with $\gamma = 7$.

Seq. \ QP	*mpb*				Δ*Time* (%)				ΔPSNR (dB)	ΔBits (%)
	28	32	36	40	28	32	36	40		
Soccer_4cif	4.10	3.06	2.05	1.36	-21.6	-27.7	-34.3	-39.3	-0.034	0.90
Container_cif	3.65	2.83	2.21	1.86	-26.5	-32.1	-37.0	-39.6	-0.044	1.19
Bus_cif	4.98	4.48	3.71	2.94	-16.9	-19.2	-23.3	-27.6	-0.017	0.36
Football_cif	4.75	3.95	3.10	2.34	-16.6	-20.9	-25.9	-30.9	-0.062	1.31
Tempete_cif	5.04	4.69	4.26	3.59	-14.8	-16.9	-19.6	-23.7	-0.023	0.55
City_4cif	4.75	4.00	3.21	2.32	-18.1	-22.3	-26.6	-32.2	-0.021	0.61
Crew_4cif	3.42	2.63	1.92	1.35	-24.4	-29.5	-34.7	-39.1	-0.040	1.11
Soccer_cif	4.35	3.25	2.26	1.67	-19.2	-25.2	-31.6	-35.9	-0.039	1.02
Harbour_4cif	4.49	4.10	3.50	2.70	-21.3	-23.2	-26.4	-38.1	-0.015	0.40
Stefan_cif	4.42	4.22	3.94	3.55	-17.6	-18.9	-20.8	-23.3	-0.034	0.67
Foreman_cif	4.19	3.58	2.99	2.41	-20.2	-23.7	-27.8	-31.7	-0.031	0.74
News_cif	3.20	2.86	2.53	2.19	-27.7	-29.8	-31.8	-34.3	-0.078	1.39
Carphone_cif	3.31	2.82	2.37	1.96	-28.9	-32.5	-35.6	-38.4	-0.031	0.72

Table 3 demonstrate results for IPPP sequences with $\gamma = 7$. In these experiments, MV search range equals to 16 pixels, I-frame period equals to 50, and Simplified UMHexagonS method is used for FME. Since inter-prediction dominates in P frames, the loss in coding performance is very unapparent. Fig.5 (c) and (d) present two examples. Because γ is smaller, only 1.35 - 5.04 modes per 4x4 block are checked on average. Although motion estimation takes a large partition of encoding time, the fast algorithm still saves up to 39.3% encoding time.



(a) (b)

(c) (d)

Fig.5. RD performance comparisons. (a) and (b) are results for all I-frames sequences. (c) and (d) are results for IPPP sequences.

5. CONCLUSION

In this paper, a novel context-based adaptive fast Intra_4x4 prediction decision algorithm is presented. By introducing a 3-order Markov random field model, the algorithm takes almost full advantage of the correlation among neighboring blocks. By using linear regression analysis, the algorithm establishes a relationship between QPs and thresholds for early terminations. Moreover, by adopting an updating process, the algorithm can adapt itself well to a variety of video sequences.

6. REFERENCES

[1] ITU-T, “Recommendation H.264,” http://www.itu.int/rec/T-REC-H.264-200503-I/en, Mar. 2005.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, “Overview of the H.264/AVC Video Coding Standard,” *IEEE Transactions on Circuits and System for Video Technology*, vol. 7, pp. 1-19, Jul. 2003.

[3] G. J. Sullivan, T. Wiegand, “Rate-Distortion Optimization for Video Compression,” *IEEE Signal Processing Magazine*, vol. 15, pp. 74-90, Nov. 1998.

[4] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. Li, D. Wu, and S. Wu, “Fast mode decision algorithm for intra-prediction in H.264/AVC video coding,” *IEEE Transactions on Circuits and System for Video Technology*, vol. 15, pp. 813- 822, Jul. 2003.

[5] B. Meng, O. C. Au, C. W. Wong, and H. K. Lam, “Efficient Intra-prediction Mode Selection for 4×4 Blocks in H.264,” *IEEE Inter. Conf. on Multi. and Expo.*, vol. 3, pp. 521-524, Jul. 2003.

[6] C. Kim, H. Shih, C.-C. J. Kuo, “Fast H.264 Intra-Prediction Mode Selection Using Joint Spatial and Transform Domain Features,” *Journal of Visual Communication and Image Representation*, vol. 17, pp. 291-310, Apr. 2006.

[7] Joint Video Team (JVT), reference software JM 12.2, http://iphome.hhi.de/suehring/tml/download/jm12.2.zip.

[8] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” VCEG-M33, 13th meeting, TX, Apr. 2001.