A Novel Bit Rate Reduction Method of H.264/AVC Intra Coding

Mohammed Golam Sarwer, and Q. M. Jonathan Wu
Department of Electrical and Computer Engineering
University of Windsor
Windsor, ON, Canada

***Abstract*— H.264/AVC intra encoder uses nine prediction modes in 4x4 and 8x8 block unit to reduce the spatial redundancies. Too many intra modes not only increase the encoder complexity but also increase the number of overhead bits. In order to reduce the number of overhead bits and computational cost, an intra mode bit rate reduction (BRR) method is proposed in this paper. In the proposed method, the numbers of prediction modes for each 4x4 and 8x8 block are selected adaptively. Based on the similarities of the reference pixels, each block is classified as one of three categories. This paper also estimates the most probable mode (MPM) from the prediction mode direction of neighboring blocks which have different weights according to their positions. Experimental results confirm that the proposed method saves 12.4% bit rate, improves the video quality by 0.37 dB on average, and requires 37.5% less computations than H.264/AVC intra encoder.**

Keywords-Video coding; intra prediction; mean absolute deviation; bit rate reduction; most probable mode

I. INTRODUCTION

H.264/AVC is the latest international video coding standard developed by ITU-T Video Coding Expert Group and the ISO/IEC Moving Picture Expert Group, which provides gains in compression efficiency of about 40% compared to previous standards [1, 2]. H.264 intra prediction offers nine prediction modes for 4x4 luma blocks, nine prediction modes for 8x8 luma blocks and four prediction modes for 16 x 16 luma blocks. However, the intra frame requires much larger bits than the inter frames, which results in buffer control difficulties and/or dropping of several frames after the intra frames in the real-time video.

A number of researchers have proposed methods to improve the rate-distortion (RD) performance of intra coder [3-6]. To reduce the intra mode bits, methods for estimating the MPM are presented in [5] and [6]. An intra coding method based on adaptive single multiple prediction is proposed [7]. In this method, each 4x4 block uses either DC prediction only or all of the nine prediction modes. The rest of this paper is organized as follows. Section II provides the review of intra-prediction method of H.264/AVC. In Section III, we describe the proposed method. The experimental results are presented in Section IV. Finally, section V concludes the paper.

II. INTRA PREDICTION OF H.264/AVC

In case of 4x4 luma block, the prediction block is calculated based on the reconstructed pixels labeled P_0-P_{12} as shown in Fig. 1 (a). For correctness, 13 reference pixels are denoted by P_0 to P_{12} and pixels to be predicted are denoted by *a* to *p*. Mode 2 is called DC prediction in which all pixels (labeled *a* to *p*) are predicted by $(P_1+P_2+P_3+P_4+P_5+P_6+P_7+P_8)/8$. The remaining modes are defined according to the different directions as shown in Fig. 1 (b). The reconstructed reference pixels and pixels to be predicted of a 8x8 block are shown in Fig. 1 (c). The directional pattern of a 8x8 block is exactly same as that of a 4x4 block which is shown in Fig. 1 (b). In addition to 4x4 and 8x8 prediction, 4 additional prediction modes (vertical, horizontal, DC and Plane) are also supported for a 16x16 block.

H.264/AVC encoder determines the best mode that meets the best RD tradeoff using RD optimization (RDO) mode decision scheme. The best mode is not directly encoded into the compressed bit stream. For current block, a mode is predicted based on the modes of upper and left blocks. This mode is defined as most probable mode (MPM).

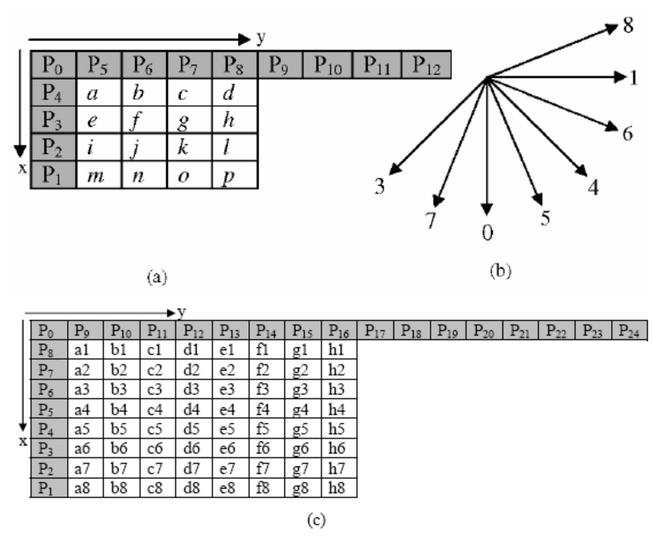

Figure 1. (a) Prediction samples of a 4x4 block (b) direction of prediction modes of 4x4 and 8x8 blocks (c) prediction samples of a 8x8 block

If the left neighboring block or the up neighboring block is unavailable, the MPM is set to 2(DC). Otherwise, the MPM is set to the minimum of the prediction mode of left block and up block. If the MPM is the same as the prediction mode, the flag is set to “1” and only one bit is needed to signal the prediction mode. When the MPM and prediction mode is different, the flag is set to “0” and additional 3 bits are required to signal the intra prediction mode.

III. Proposed Bit Rate Reduction (BRR) Method

A. *Adaptive number of modes*

Although H.264/AVC intra coding method provides good compression ratio, owing to the use of nine prediction modes of 4x4 and 8x8 luma blocks, its computational complexity increases drastically. Using nine prediction modes in intra 4x4 and 8x8 block unit for a 16x16 MB can reduce the spatial redundancies, but it may needs a lot of overhead bits to represent the prediction mode of each 4x4 or 8x8 block. Based on the variation of neighboring pixels, the proposed method classifies a block as one of three different cases.

N	N	N	N	N	N	N	N	N
N	*a*	*b*	*c*	*d*				
N	*e*	*f*	*g*	*h*				
N	*i*	*j*	*k*	*l*				
N	*m*	*n*	*o*	*p*				

Figure 2. Case 1: All of the reference pixels have similar value

P_0	N	N	N	N	N	N	N	N
P_4	*a*	*b*	*c*	*d*				
P_3	*e*	*f*	*g*	*h*				
P_2	*i*	*j*	*k*	*l*				
P_1	*m*	*n*	*o*	*p*				

Figure 3. Case 2: The reference pixels of up and up-right block have similar value

1) *Case 1*

As shown in Fig. 2, if all of the reference pixels are same, the prediction values of all of the nine directional modes are same. In this case, it does not need to calculate the entire prediction modes. Only DC mode can be used, so that the prediction mode bit can be skipped. If mean absolute deviation (MAD) σ_1 of all of the neighboring pixels is less than the threshold T_1, only DC prediction mode is used. The MAD σ_1 and mean μ_1 is defined as,

$$\sigma_1 = \sum_{i=1}^{12} \left| P_i - \mu_1 \right| , \quad \mu_1 = [\sum_{i=1}^{4} P_i + (\sum_{i=5}^{8} P_i) << 1 + \sum_{i=9}^{12} P_i] >> 4 \quad (1)$$

where << and >> means shift left and shift right operator. P_i is the i-th reference pixel of Fig. 1(a) and μ_1 is the weighted mean value of block boundary pixels. In order to set the threshold T_1, we have done several experiments for four different types of video sequences (*Mother & Daughter, Foreman, Bus* and *Stefan*) with CIF format at different quantization parameters (QPs). By changing the threshold, we observed the RD performance. We found that threshold T_1 is independent on the type of video sequence but depends on the QP values. The threshold T_1 is approximated as follows:

$$T_1 = \begin{cases} QP + 12 & \text{if } QP \le 24 \\ 5QP - 90 & \text{Otherwise} \end{cases} \quad (2)$$

2) *Case 2*

As shown in Fig. 3, if all of the reference pixels of up and up-right blocks are same, vertical, diagonal-down-left, vertical-left, vertical-right and horizontal-down modes produce the similar prediction value. That’s why, in the proposed method we have chosen only vertical prediction mode from this group. If MAD σ_2 of the neighboring pixels of up and up-right blocks is less than the threshold T_2, four prediction modes (vertical, horizontal, diagonal-down-right and horizontal-up) are used. Instead of using 3 bits of original encoder, each of four prediction modes is represented by 2 bits. Vertical, horizontal, diagonal-down-right and horizontal-up mode are represented by “00”, “01”, “10” and “11”, respectively. Threshold T_2 is selected as same way of T_1. T_2 also depends on the QP and better results were found at $T_2 = (2T_1 / 3)$. The MAD σ_2 and mean μ_2 are defined as,

$$\sigma_2 = \sum_{i=5}^{12} \left| P_i - \mu_2 \right| , \text{ and } \mu_2 = (\sum_{i=5}^{12} P_i) >> 3 \quad (3)$$

where μ_2 is the mean value of block boundary pixels of top and top-right blocks.

The flow diagram of the proposed method is presented in Fig. 4. If $\sigma_1 < T_1$, only DC prediction mode is used. In this case computational expensive RDO process is skipped. In addition, no bit is necessary to represent intra prediction mode. On the other hand, if $\sigma_1 < T_1$ is not satisfied, encoder calculates the MAD σ_2 and threshold T_2. If $\sigma_2 < T_2$, vertical, horizontal, diagonal-down-right and horizontal-up modes are used as the candidate modes of RDO process. If $\sigma_2 < T_2$ is not satisfied, most probable mode (MPM) is estimated and nine prediction modes are used as the candidate mode of the RDO process, as in H.264/AVC. In this case, either 1 or 4 bits are allocated to represent the intra prediction mode. The new prediction mode numbers are recorded in Table I. Since diagonal-down-left, vertical right, horizontal down and vertical left predictions modes are not utilized in the previous cases, the probability of these modes are very high in this case.

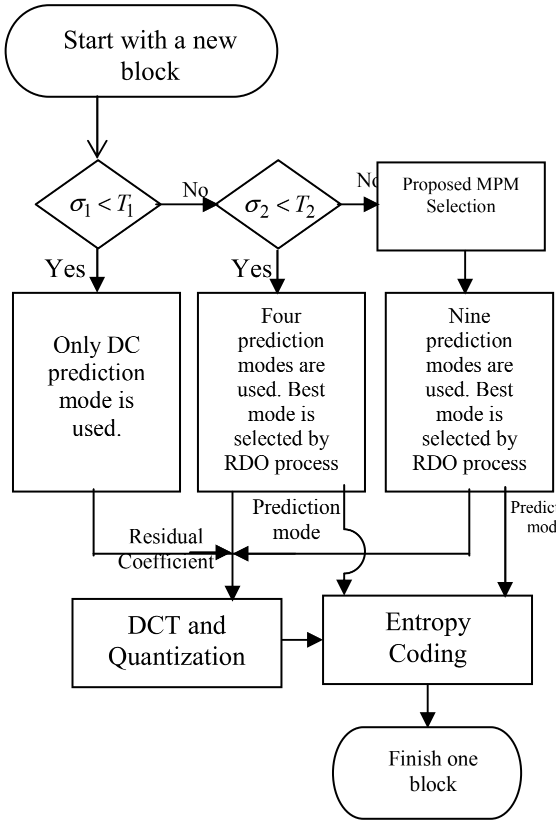

Figure 4. Flow diagram of proposed method

TABLE I. PREDICTION MODES RECORDING OF THE PROPOSED METHOD

Mode	Mode number H.264/AVC	Mode number Proposed
Diagonal-down-left	3	0
Vertical-right	5	1
Horizontal-down	6	2
Vertical-left	7	3
Vertical	0	4
Horizontal	1	5
DC	2	6
Diagonal-down-right	4	7
Horizontal-up	8	8

Since 8x8 intra prediction also uses 9 prediction modes, the proposed method is also applied to 8x8 intra prediction mode. Assume P_i is the i-th reconstructed pixel of Fig. 1 (c). Mean absolute Deviations (MADs) and thresholds of a 8x8 block are defied as

$$\mu_{1_{8x8}} = (\sum_{i=1}^{8} P_i + (\sum_{i=9}^{16} P_i) << 1 + \sum_{i=17}^{24} P_i) >> 5$$

$$\text{and } \sigma_{1_{8x8}} = \sum_{i=1}^{24} \left| P_i - \mu_{1_{8x8}} \right| \quad (4)$$

$$\mu_{2_{8x8}} = (\sum_{i=9}^{24} P_i) >> 4 \text{ and } \sigma_{2_{8x8}} = \sum_{i=9}^{24} \left| P_i - \mu_{2_{8x8}} \right| \quad (5)$$

$$T_{1_{8x8}} = \begin{cases} 2QP + 24 & \text{if } QP \le 24 \\ 10QP - 180 & \text{Otherwise} \end{cases}$$

$$\text{and } T_{2_{8x8}} = 2T_{1_{8x8}} / 3 \quad (6)$$

From some simulations, we have found that a significant number of blocks still calculate 9 prediction modes. If the MPM is the best mode, only 1 bit is used; otherwise 4 bits are required to represent the prediction mode. Therefore, if we can develop a more accurate method to estimate the MPM, a significant percentage of blocks will use only 1 bit for mode information.

B. Selection of MPM

Natural video sequences contain a lot of edges and these edges are usually continuous. If the direction from the neighboring block to the current block is identical to the prediction mode direction of the neighboring block, there is high possibility that the best prediction mode of current block is also identical to be the prediction mode direction. So, the weight of the proposed method is proportional to the absolute difference between block direction and mode directions. The mode direction (θ_m) is calculated by considering horizontal mode as base and tabulated in Table II.

TABLE II. MODE DIRECTION (θ_m)

Mode	Direction
Vertical	$\pi / 2$
Horizontal	0
Diagonal-down-left	$3\pi / 4$
Diagonal-down-right	$\pi / 4$
Vertical- right	$3\pi / 8$
Horizontal-down	$\pi / 8$
Vertical-left	$5\pi / 8$
Horizontal-up	$-\pi / 8$

The block direction (θ_B) and block distance (D_B) are calculated by following set of equations.

$$\theta_B = \tan^{-1} \frac{y_c - y_n}{x_c - x_n} \quad (7)$$

$$D_B = \left| y_c - y_n \right| + \left| x_c - x_n \right| \quad (8)$$

Where, (x_c, y_c) and (x_n, y_n) are the position of current and neighboring block, respectively. The mode of the neighboring block is denoted as M_n . If the distance between current and neighboring block is higher, the correlations between the blocks are lower and weights are also lower too. Based on these observations weight of neighboring mode M_n is defined as

$$W(M_n) = \min[0, \frac{\alpha}{D_B} - \beta|\theta_B - \theta_m|] \quad (9)$$

where α and β are the proportionally constant. Based on simulation, α and β are selected as 6 and $8/\pi$.

For prediction mode of each neighboring block weight is calculated and updated by adding with the weight of same mode. Since, DC has no unified direction, if the neighboring mode is DC, weight corresponding to this block is set to 0. The weight of each prediction mode is counted up and find out the mode with highest weight $W_{\max}$. If the maximum weight $W_{\max}$ is very low, it seems that there is no continuation of edges. In this case, possibility of DC prediction mode to be the best mode is higher. If maximum weight $W_{\max}$ is less than a threshold T_{MPM}, the MPM is the DC mode; otherwise the MPM is the mode with maximum weight $W_{\max}$. Following is the step by step algorithm of the proposed method.

Step 1: Initialize weight (W) of each mode to zero.

Step 2:

For each of four neighboring blocks (up, left, up-left, up-right),

If neighboring mode $M_n = DC$, $W(M_n) \mathrel{+}= 0$.

Otherwise

(a) calculate block direction, θ_B and D_B

(b) Find mode direction of the neighboring mode M_n from Table II.

(c) Calculate weight of neighboring mode: $W(M_n) \mathrel{+}= \min[0, \frac{\alpha}{D_B} - \beta|\theta_B - \theta_m|]$

End of block

Step 3: Find the maximum weight $W_{\max}$ and the mode that has maximum weight.

Step 4: If maximum weight $W_{\max}$ is less than T_{MPM}, the most probable mode is the DC mode; otherwise MPM is the mode with maximum weight $W_{\max}$.

In order to find the threshold T_{MPM}, we have done some simulations. Four different types of video sequences (Mother & Daughter, Foreman, Bus and Stefan) were encoded by changing the value of TMPM from 1 to 10 and RD performances were observed. Better results were found at $T_{MPM} = 5$.

TABLE III. RD PERFORMANCE WITH 4X4 MODE ONLY

Sequences	Method in [7]		Proposed ANM only		Proposed BRR (ANM+MPM)	
	Δ PSNR	Δ Rate %	Δ PSNR	Δ Rate %	Δ PSNR	Δ Rate%
Grand Mother (QCIF)	0.37	-15.4	0.41	-16.4	0.42	-17.0
Salesman (QCIF)	0.32	-12.9	0.39	-13.5	0.40	-14.6
Stefan (QCIF)	0.10	-2.7	0.19	-5.87	0.20	-6.0
Container (QCIF)	0.09	-3.1	0.15	-5.35	0.18	-6.7
Car phone (QCIF)	0.66	-18.4	0.79	-22.3	0.83	-23.8
Silent (CIF)	0.35	-15.4	0.40	-17.3	0.42	-18.0
Bus (CIF)	0.11	-3.8	0.12	-4.58	0.15	-4.1
Hall (CIF)	0.32	-8.6	0.37	-9.87	0.42	-11.3
Mobile Calendar (HD-1280x720)	0.19	-6.8	0.25	-9.31	0.27	-9.8
Average	0.28	-9.7	0.34	-11.6	0.37	-12.4

TABLE IV. COMPLEXITY PERFORMANCE WITH 4X4 MODE ONLY

Sequences	Method in [7]		Proposed ANM only		Proposed BRR (ANM+MPM)	
	ΔT_1 %	ΔT_2 %	ΔT_1 %	ΔT_2 %	ΔT_1 %	ΔT_2 %
Grand Mother (QCIF)	-39.7	2.09	-52.1	1.99	-49.1	2.06
Salesman (QCIF)	-31.2	1.19	-37.9	0.91	-35.7	1.32
Stefan (QCIF)	-17.9	0.39	-25.4	0.33	-22.6	0.65
Container (QCIF)	-31.3	0.45	-39.5	0.41	-37.9	0.67
Car phone (QCIF)	-33.8	0.42	-46.0	0.39	-42.0	0.55
Silent (CIF)	-35.8	1.74	-45.8	1.53	-43.0	2.02
Bus (CIF)	-16.4	0.99	-32.0	0.82	-28.8	1.67
Hall (CIF)	-38.8	1.45	-48.5	1.18	-45.0	1.87
Mobile Calendar (HD-1280x720)	-27.6	1.55	-34.7	1.59	-33.0	3.01
Average	-30.3	1.14	-40.2	1.01	-37.5	1.53

IV. SIMULATION RESULTS

To evaluate the performance of the proposed method, JM 12.4 [8] reference software is used in simulation. Different types of video sequences with different resolutions are used as test materials. Simulation conditions are (a) QPs are 28, 36, 40, 44 (b) entropy coding: CABAC (c) RDO on (d) frame rate: 30 fps and (e) number of frames: 100. The comparison results are produced and tabulated based on the average difference in the total encoding (ΔT_1 %) and decoding (ΔT_2 %) time, the average PSNR differences ($\Delta PSNR$), and the average bit rate difference ($\Delta R\%$). PSNR and bit rate differences are calculated according to the numerical averages between RD curves [9]. The encoding (ΔT_1 %) and decoding (ΔT_2 %) complexity is measured as follows

$$\Delta T_1\% = \frac{T_{p_{enc}} - T_{o_{enc}}}{T_{o_{enc}}} \times 100\% \quad (10)$$

$$\Delta T_2\% = \frac{T_{p_{dec}} - T_{o_{dec}}}{T_{o_{dec}}} \times 100\% \quad (11)$$

where, $T_{o_{enc}}$ and $T_{o_{dec}}$ are the total encoding and decoding time of the JM 12.4, respectively. $T_{p_{enc}}$ and $T_{p_{dec}}$ are the total encoding and decoding time of the proposed method, respectively.

A. Experiments with 4x4 intra mode only

In this experiment all frames are intra coded only 4x4 mode is enabled. The performance comparisons are presented in Table III and IV. In these tables, a positive value indicates increment and a negative value represents decrement. In case of method in [7], the average PSNR improvement is about 0.28 dB and average bit rate reduction is about 9.7%. Whereas in our proposed ANM only method, the average PSNR improvement is about 0.34 db and average bit rate reduction is about 11.6%. The proposed ANM only method also reduces the computation of the original encoder by 40%. Although this method introduces some extra computations of the decoder side, the simulation results of Table IV confirm that, the computational overhead of the decoder is very low (about 1.01%). If we combine our ANM and MPM method together, about 12.4% bit rate reduction is achieved along with a 0.37 dB improvement in PSNR in the expense of 1.53% increment of decoding time.

TABLE V. PERFORMACNE RESULTS OF BBR WITH ALL INTRA MODES

Sequence	$\Delta PSNR$ in dB	ΔR %	ΔT_1 %	ΔT_2 %
Grand Mother (QCIF)	0.34	-14.5	-38.2	3.89
Sales man (QCIF)	0.40	-13.8	-29.2	1.79
Stefan (QCIF)	0.18	-5.6	-20.9	1.20
Container (QCIF)	0.12	-3.9	-34.1	1.64
Carphone (QCIF)	0.75	-17.2	-36.1	2.97
Silent (CIF)	0.31	-10.9	-36.0	2.99
Bus (CIF)	0.15	-4.8	-21.4	3.21
Hall (CIF)	0.30	-9.1	-36.7	1.01
Mobile Calendar (HD-1280x720)	0.20	-11.6	-30.8	1.88
Average	0.31	-10.2	-31.5	2.29

B. Experiments with all intra modes

In this experiment all 100 frames are encoded by intra coding and all intra modes (4x4, 8x8, and 16x16) are enabled. The results are tabulated in Table V. Here proposed BBR method is implemented in 4x4 and 8x8 blocks. Since only small amount of MBs are encoded with 16x16 modes, the proposed methods are not implemented in 16x16 mode for computational difficulties. We have seen that the average gain is in the range of 0.31 dB PSNR and 10.2% bit rate saving, with a maximum for sequence *Carphone* with 0.75 dB and 17.2%. We have seen that the proposed method reduces 31.5% computation of original encoder. The computation increment of decoder side is very low and that is 2.29% on average. The rate-distortion (RD) curves of *Carphone* video sequence are plotted in Fig. 5 which shows that our proposed method is superior to that of the original H.264/AVC encoder.

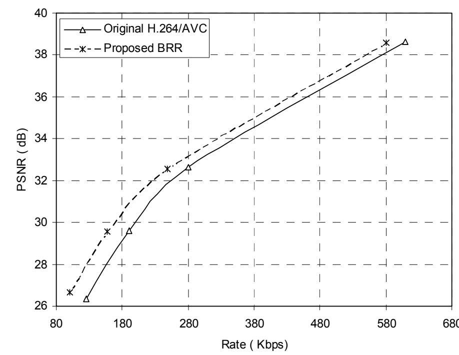

Figure 5. RD curve of carphone video sequence

V. CONCLUSION

In this paper, an intra mode bit rate reduction (BRR) scheme for representing the intra prediction mode is described. In proposed BRR method the number of prediction modes of a block is selected based on the MAD of the neighboring pixels. Most of the blocks spent either 1 or 2 bits to encode the intra prediction mode information. An efficient technique for selecting the most probable mode is also developed based on the continuity of the edge direction. The proposed method not only improves the RD performance but also reduces the computational complexity of H.264/AVC intra coder.

REFERENCES

[1] ISO/IEC 14496-10, Information Technology-Coding of audio-visual objects-Part:10: Advanced Video Coding. ISO/IEC JTC1/SC29/WG11 (2004)

[2] T. Weigand , G. Sullivan, G. Bjontegaard, and A. Luthra " Overview of H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Syst. Video Technol.,* vol. 13, July 2003, pp 560-576.

[3] Y. Ye, and M. Karczewicz, " Improved H.264 Intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning", in Proc. *IEEE ICIP 2008*, pp. 2116-2119.

[4] L. Wang, L. M. Po, Y. M. S. Uddin, K. M. Wong, and S. Li, " A novel weighted cross prediction for H.264 intra coding", *IEEE ICME 2009*, pp. 165-168.

[5] D. Y. Kim, D. K. Kim, and Y. L. Lee, "A New Method for Estimating Intra Prediction Mode in H.264/AVC", *IEICE Trans. Fund.s of Electron., Commun. and Computer Scienc.*, vol. E91-A, june 2008, pp. 1529-1532.

[6] J. Lee, J. S. Choi, J. Hong, and H.l Choi, Intra-mixture Prediction Mode and Enhanced Most Probable Mode Estimation for Intra Coding in H.264/AVC, *Fifth International Joint Conference on INC, IMS and IDC, 2009*, NCM '09, August 2009, pp. 1619-1622.

[7] D. Y. Kim, K. H. Han, and Y. L. Lee, Adaptive single-multiple prediction of H.264/AVC intra coding, *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 20, no. 4, April 2010, pp. 610-615.

[8] JM reference software 12.4, http://iphome.hhi.de/suehring/tml/download/old_jm/jm12.4.zip

[9] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," presented at the 13th VCEG-M33 Meeting, Austin, TX, April 2001