Enhanced Intra Coding of H.264/AVC Advanced Video Coding Standard with Adaptive Number of Modes

Mohammed Golam Sarwer and Q.M. Jonathan Wu

Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, N9B 3P4, Canada
{sarwer,jwu}@uwindsor.ca

Abstract. In H.264/AVC intra coding, DC mode is used to predict the regions with no unified direction and the predicted values of all pixels are same. Therefore, the smoothly varying regions are not well de-correlated. In order to address this issue, this paper proposes an improved DC prediction mode based on the distance between the predicted and reference pixels. On the other hand, using the nine prediction modes in intra 4x4 and 8x8 block unit can reduce the spatial redundancies, but it needs a lot of overhead bits. In order to reduce the number of overhead bits and computational cost of the encoder, this paper adaptively selects the number of prediction mode for each 4x4 or 8x8 block. Experimental results confirm that the proposed methods save 14.8% bit rate and improve the video quality by 0.44 dB on average. The proposed method saves about 37.8% computation of the H.264/AVC intra coding method.

Keywords: H.264/AVC, intra coding, DC prediction, mode, variance.

1 Introduction

H.264/AVC is the newest international video coding standard developed by the Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG [1]. The rate-distortion (RD) performance of H.264/AVC is superior to any other conventional video coding standard. H.264/AVC offers a rich set of prediction patterns for intra prediction, i.e. nine prediction modes for 4x4 luminance (luma) blocks, nine prediction modes for 8x8 luma blocks and four prediction modes for 16 x 16 luma blocks. However, the RD performance of the intra frame coding is still lower than that of inter frame coding. Thus the development of efficient intra coding is important not only for the overall bit rate reduction but also for the efficient streaming.

A block based extra/inter-polation method by changing the sub-block coding order in a Macroblock (MB) is proposed in [2]. Since three additional directional predictions are utilized, the computational complexity of intra prediction increases drastically. In addition to that number of overhead bits to represent the prediction mode also increases. A simplified Bi-directional intra prediction (BIP) [3] method combines two existing prediction modes to form bi-directional prediction modes. In order to improve the performance of DC prediction, a distance based weighted prediction method (DWP) based on the distance of predicted and reference pixels is introduced

A. An et al. (Eds.): AMT 2010, LNCS 6335, pp. 361–372, 2010.

in [4]. A fast algorithm of DWP is developed in [5]. However, from simulation it is shown that improvement of RD performance of these methods is marginal. A pixel based differential intra coding method is developed in [15] but this method is only suitable for lossless intra coding.

H.264/AVC uses rate-distortion optimization (RDO) technique to get the best coding mode out of nine prediction modes in terms of maximizing coding quality and minimizing bit rates. A lot of algorithms are developed to reduce the computation of H.264/AVC encoder but all of these methods sacrifice video quality [9-14]. Using the nine prediction modes in intra 4x4 and 8x8 block unit can reduce the spatial redundancies, but it may need a lot of overhead bit to represent the prediction mode of each 4x4 or 8x8 block. A new intra coding method based on adaptive single-multiple prediction (SMP) is proposed in order to reduce not only the overhead mode bits but also computational cost [6]. If the variance of the neighboring pixels of upper and left blocks is less than the threshold, only DC prediction is used and that does not need to prediction mode bits. Otherwise nine prediction modes are computed. But the reference pixels in up-right blocks are not consider during early termination. If variance of reference pixels of upper and left blocks is very low and variance of upright blocks is higher, all of the prediction modes except diagonal down left and vertical left modes are similar. In this case, only DC prediction mode is not enough to maintain good PSNR and compression ratio. Therefore, still there are some rooms to improve the performance of intra encoder. The goal of this paper is development of efficient DC prediction method suitable for smooth image region and reduction of the overhead bits to represent intra prediction modes.

The remainder of this paper is organized as follows. Section 2 provides the review of intra-prediction method of H.264/AVC. In Section 3, we describe the proposed methods. The experimental results are presented in Section 4. Finally, section 5 concludes the paper.

2 Intra Prediction of H.264/AVC

For the luma samples, intra prediction may be formed for each 4x4 block or for each 8x8 block or for a 16x16 macroblock. There are a total of 9 optional prediction modes for each 4x4 and 8x8 luma block; 4 optional modes for a 16x16 luma block. Similarly for chroma 8x8 block, another 4 prediction directions are used. The prediction of a 4x4 block is computed based on the reconstructed samples labeled P_0-P_{12} as shown in Fig. 1 (a). The grey pixels (P_0-P_{12}) are reconstructed previously and considered as reference pixels of the current block. For correctness, 13 reference pixels of a 4x4 block are denoted by P_0 to P_{12} and pixels to be predicted are denoted by P_{13} to P_{28}. Mode 2 is called DC prediction in which all pixels (labeled P_{13} to P_{28}) are predicted by $(P_1+P_2+P_3+P_4+P_5+P_6+P_7+P_8+4)/8$ and mode 0 specifies the vertical prediction mode in which pixels (labeled P_{13}, P_{14}, P_{15} and P_{16}) are predicted from P_5, and the pixels (labeled P_{17}, P_{22}, P_{21}, and P_{20}) are predicted from P_6, and so on. The remaining modes are defined similarly 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. The best mode is the one that has the minimum RD cost and this cost is expressed as,

$$J_{RD} = SSD + \lambda \cdot R \tag{1}$$

where, SSD is the sum of squared difference between the original block and the reconstructed block . R is the true bits needed to encode the block and λ is an exponential function of the quantization parameter (QP).

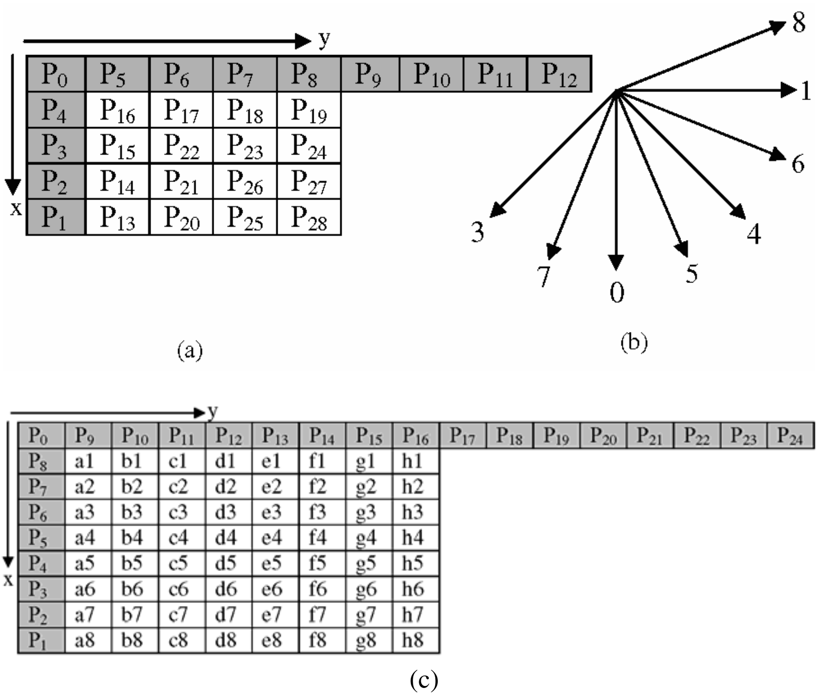


Fig. 1. (a) Prediction samples of a 4x4 block (b) direction of prediction modes of 4x4 and 8x8 blocks (c) prediction samples of a 8x8 block

3 Proposed Enhanced Intra Coding

3.1 Improved DC Prediction for 4x4 Block

In H.264/AVC, DC mode is used to predict regions with no unified direction and the predicted values of all pixels are same. But the correlation that exists between predicted pixels and reference pixels are not absolutely considered to predict the DC prediction mode. Thus, the prediction signal generated by DC prediction is not well matched to the original signal, and a large number of bits is required for encoding the difference between the predicted and original signal. In order to improve the performance of DC prediction of a 4x4 block, this section proposes a distance based prediction based on the distance between reference and current pixels. It is well known that Gaussian-like distribution can approximate local intensity variations in smooth image region. Therefore, the correlation between neighboring pixels would be attenuated while the distance is increased and negligible when pixels are far apart. Therefore, prediction accuracy is degraded if all of the pixels of a 4x4 block are predicted from the same reference pixels.

Let us consider a 4x4 block as shown in Fig. 1 (a). In original DC prediction of H.264/AVC all of the pixels P_{13} to P_{28} of the 4x4 block are predicted from the neighboring reconstructed pixels P_1 to P_8. In the proposed IDCP method, pixels P_{13} to P_{19} are predicted from reconstructed pixels P_1 to P_8, P_{20} to P_{24} are predicted from previously predicted pixels P_{13} to P_{19}, P_{25} to P_{27} are calculated from pixels P_{20} to P_{24} and the reference pixels of P_{28} are P_{25} to P_{27}. The predicted pixels P_r are estimated by following equation.

$$P_r = \left\lfloor \left(\sum_{i=m}^{n} W_i P_i\right) / \left(\sum_{i=m}^{n} W_i\right) \right\rfloor, \text{ where } r = 13 \text{ to } 28. \tag{2}$$

Here P_r is the pixel to be predicted and P_i is the reference pixel and $\lfloor\ \rfloor$ is the truncation operation to generate an integer value. Table 1 show the value of m and n for different predicted pixels r and W_i is the weight of i-th reference pixel. If the distance between predicted pixels P_r and reference pixels P_i is lower, the correlation between P_r and P_i is higher, and W_i is also higher. Therefore, weighting factor W_i is inversely proportional to the distance and defined as follows.

$$W_i = \lfloor 8 / D \rfloor \tag{3}$$

where D is the distance between reference pixel and pixel to be predicted and defined as $D = \left|B_{r,x} - B_{i,x}\right| + \left|B_{r,y} - B_{i,y}\right|$. $B_{r,x}$ and $B_{r,y}$ are the x and y positions of the predicted pixel P_r and similarly $B_{i,x}$ and $B_{i,y}$ are the x and y positions of the reference pixel P_i. In our method, the predicted value of the neighboring pixels obtained in the previous step is used to predict current pixels. Since some of the predicted values are based on other predicted values, the order of the calculation of predicted values are P_{13} to P_{28} as shown in four steps in Table 1.

Table 1. Value of m and n of (2) with different predicted pixels

Step	Predicted pixels r	Reference pixels i	m	n
Step 1	r= 13 to 19	i= 1 to 8	1	8
Step 2	r= 20 to 24	i= 13 to 19	13	19
Step 3	r = 25 to 27	i= 20 to 24	20	24
Step 4	r= 28	i= 25 to 27	25	27

Let us assume, we would like predict pixel P_{26} as shown in Fig. 1 (a). According to third row of Table 1, in order to predict P_{26}, the reference pixels are P_{20}, P_{21}, P_{22}, P_{23}, and P_{24}. By substituting the value of r, m and n in (2) we get

$$P_{26} = \left\lfloor \frac{W_{20}P_{20} + W_{21}P_{21} + W_{22}P_{22} + W_{23}P_{23} + W_{24}P_{24}}{W_{20} + W_{21} + W_{22} + W_{23} + W_{24}} \right\rfloor \quad (4)$$

Consider that the position of predicted pixel P_{26} is $(B_{26,x}, B_{26,y})$ and therefore, as shown in Fig. 1(a) the position of reference pixels $(B_{i,x}, B_{i,y})$, i=20 to 24, are $(B_{26,x}+1, B_{26,y}-1)$, $(B_{26,x}, B_{26,y}-1)$, $(B_{26,x}-1, B_{26,y}-1)$, $(B_{26,x}-1, B_{26,y})$ and $(B_{26,x}-1, B_{26,y}+1)$, respectively . By substituting these values in (3), we can calculate the weighting factors W_{20} to W_{24}. It is shown that in order to calculate (3), computationally expensive division operation is necessary. In order to lessen computations, W_i is simplified as follows.

$$W_i = 2^{K_i} \quad \text{where} \quad K_i = \begin{cases} 3 & \text{if } D = 1 \\ 2 & \text{if } D = 2 \\ 1 & \text{if } D = 3 \text{ or } 4 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

By substituting (5) into (2) and replacing multiplication operation by shifting,

$$P_r = \left\lfloor \frac{(\sum_{i=m}^{n} P_i << K_i)}{\sum_{i=m}^{n} W_i} \right\rfloor \text{, where } r = 13 \text{ to } 28 \quad (6)$$

The division and truncation operation of (6) is inconvenient for hardware implementation. We have seen that for a particular value of r, the denominator of (6) is constant throughout the encoding and decoding process and independent on the intensity of the pixels. In order to avoid computational expensive division operator, (6) is approximated as follows:

$$P_r = [(\sum_{i=m}^{n} P_i << K_i) + C_r \times P_{r,left}] >> 5 \quad (7)$$

where $C_r = 2^5 - \sum_{i=m}^{n} W_i$ and value of C_r is given in Table 2. $P_{r,left}$ is the immediate left pixel of P_r. For example, $P_{r,left} = P_{15}$ for r=22. Obviously, (7) is faster than (2) and hardware inconvenient division and truncation operations are avoided. According to (7), each 4x4 block needs 125 shifting, 109 addition and 16 multiplication operator for improved DC prediction.

3.2 Adaptive Number of Modes for 4x4 and 8x8 Blocks

Although H.264/AVC intra coding method provides good compression ratio, due to the use of nine prediction modes, its computational complexity is increased

Table 2. Value of C_r with different predicted pixels

r	C_r
13, 19, 20, 24	11
14, 18	8
15, 17	5
16, 22	0
21, 23	6
25, 27, 28	12
26	4

drastically. Using the 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 bit to represent the prediction mode of each 4x4 or 8x8 block. In order to reduce the number of overhead bits and computational cost, this section proposed an algorithm as given below.

3.2.1 Case 1

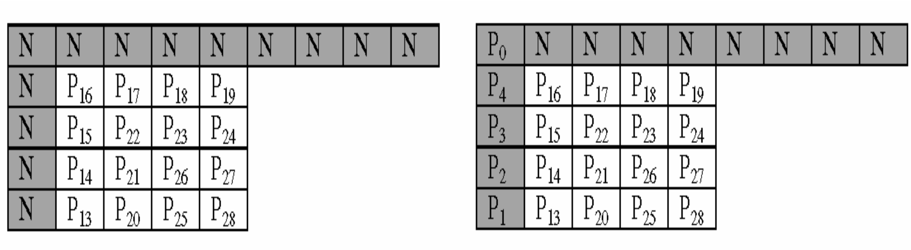
As shown in Fig. 2(a), if all reference pixels are same, the predicted values of nine directional predictions are same. In this case, it does not need to calculate the entire prediction modes. In this case, only DC mode can be used for encoder and decoder prediction and the prediction mode bit can be skipped. In order to classify the 4x4 block in this category, variance σ_1 and threshold T_1 are calculated. If variance σ_1 of all of the neighboring pixels is less than the threshold T_1, only DC prediction mode is used and it does not need the prediction mode bits. The variance σ_1 and mean μ_1 are defined as,

$$\sigma_1 = \sum_{i=1}^{12} |P_i - \mu_1| \quad \text{and} \quad \mu_1 = \left\lfloor (\sum_{i=1}^{12} P_i)/12 \right\rfloor \tag{8}$$

where P_i is the i-th pixel of Fig. 1(a) and μ_1 is the mean value of block boundary pixels. In order to avoid computational expensive division and truncation operations, μ_1 is replaced by weighted mean value (μ_1') as

$$\mu_1' = (\sum_{i=1}^{4} P_i + (\sum_{i=5}^{8} P_i) << 1 + \sum_{i=9}^{12} P_i) >> 4 \quad \text{and} \quad \sigma_1 = \sum_{i=1}^{12} \left| P_i - \mu_1' \right| \tag{9}$$

Here << and >> are the shift left and shift right operator, respectively. 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 QP values. *Mother & Daughter* represents simple and low motion video sequence. *Foreman* and *Bus* contain medium detail and represent medium motion video sequences. *Stefan* represents high detail and complex motion video sequence. By changing the threshold, we observed the RD performance and found that threshold T_1 is independent on the type of video sequence but depends on the QP values. By using the polynomial fitting technique, the threshold value T_1 is approximated as follows:



(a) (b)

Fig. 2. (a) Case 1: All of the reference pixels have similar value (b) Case 2: The reference pixels of up and up-right block have similar value

$$T_1 = \begin{cases} QP + 12 & \text{if } QP \le 24 \\ 5QP - 90 & \text{Otherwise} \end{cases} \tag{10}$$

3.2.2 Case 2

As shown in Fig. 2(b), 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 same prediction values. That's why, in the proposed method we have chosen only one mode from this set and it is the vertical prediction mode. If variance 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 and one of them is selected through RDO process. Instead of using 3 bits of original encoder, each of the four prediction mode is represented by 2 bits. In this case, the binary representations of prediction modes are recorded as shown in Table 3 and hence a significant amount of mode bits are saved. Threshold T_2 is selected in the same way as T_1. T_2 also depends on the QP and better results were obtained at

$$T_2 = \lfloor (2T_1 / 3) \rfloor \approx (T_1 >> 1 + T_1 >> 3) \tag{11}$$

The variance σ_2 and mean μ_2 are defined as,

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

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

Table 3. Binary representation of modes of case 2

Mode	Binary representation
Vertical	00
Horizontal	01
Diagonal-down-right	10
Horizontal-up	11

The flow diagram of the proposed ANM method is presented in Fig. 3. The variance σ_1 and threshold T_1 are calculated at the start of the mode decision process and if the variance is less than the threshold ($\sigma_1 < T_1$) only DC prediction mode is used. In this case computational expensive RDO process is skipped and a lot of computations are saved. In addition, no bit is necessary to represent intra prediction mode because only one mode is used. On the other hand, if $\sigma_1 < T_1$ is not satisfied, encoder calculates the variance σ_2 and threshold T_2. If $\sigma_2 < T_2$, vertical, horizontal, diagonal-down-right and horizontal-up modes are used as candidate modes in RDO process. A substantial saving in computations is achieved using 4 prediction modes instead of 9 modes of the original RDO process. In order to represent the best mode, 2 bits are sent to the decoder and Table 3 shows the four prediction modes with corresponding binary representations. As shown in Table 3, if the diagonal-down-right mode is selected as the best mode, the encoder sends "10" to the decoder. In this category, only 2 bits are used to represent the intra prediction mode whereas 3 bits are used in the original encoder. Consequently a large number of intra prediction mode bits are saved. If $\sigma_2 < T_2$ is not satisfied, nine prediction modes are used as the candidate mode and one of them is selected through the RDO process, as in H.264/AVC. The new prediction mode numbers are recorded and compared against H.264/AVC in Table 4.

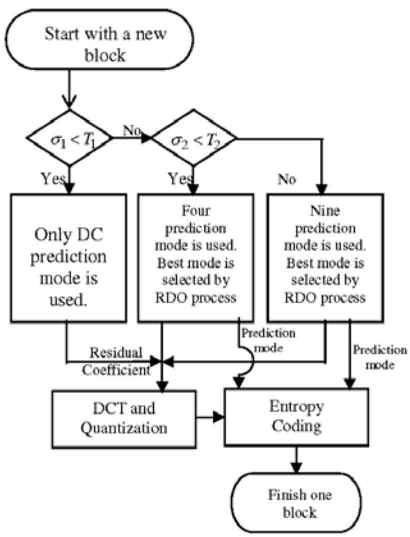


Fig. 3. Flow diagram of proposed ANM method

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

$$\mu'_{1_{8x8}} = (\sum_{i=1}^{8} P_i + (\sum_{i=9}^{16} P_i) << 1 + \sum_{i=17}^{24} P_i) >> 5 \quad \text{and} \quad \sigma_{1_{8x8}} = \sum_{i=1}^{24} \left| P_i - \mu_{1_{8x8}}' \right| \tag{13}$$

$$\mu_{2_{8x8}} = (\sum_{i=9}^{24} P_i) >> 4 \text{ and } \sigma_{2_{8x8}} = \sum_{i=9}^{24} \left| P_i - \mu_{2_{8x8}} \right| \tag{14}$$

$$T_{1_{8x8}} = \begin{cases} 2QP + 24 & \text{if } QP \le 24 \\ 10QP - 180 & \text{Otherwise} \end{cases} \quad \text{and } T_{2_{8x8}} = (T_{1_{8x8}} >> 1 + T_{1_{8x8}} >> 3) \tag{15}$$

Table 4. 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

4 Simulation Results

To evaluate the performance of the proposed method, JM 12.4 [7] reference software is used in simulation. 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 (ΔP), and the average bit rate difference ($\Delta R\%$). PSNR and bit rate differences are calculated according to the numerical averages between RD curves [8]. The encoding (ΔT_1 %) and decoding (ΔT_2 %) complexity is measured as follows

$$\Delta T_1 = \frac{T_{p_{enc}} - T_{o_{enc}}}{T_{o_{enc}}} \times 100\% \tag{16}$$

$$\Delta T_2 = \frac{T_{p_{dec}} - T_{o_{dec}}}{T_{o_{dec}}} \times 100\% \tag{17}$$

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

Table 5(a). RD performances of proposed methods (only 4x4 modes, All I frames)

	DWP[4]		SMP [6]		Proposed IDCP only		Prop ANM only		ANM + IDCP	
	Δ P	Δ*R* %	Δ P	Δ*R* %	Δ P	Δ*R* %	Δ P	Δ*R* %	Δ P	Δ*R* %
Grand Mother (QCIF)	0.04	-1.4	0.37	-15.4	0.11	-4.6	0.41	-16.4	0.47	-18.0
Sales man (QCIF)	0.02	-0.2	0.32	-12.9	0.12	-6.6	0.39	-13.5	0.42	-18.9
Stefan (QCIF)	0.01	-0.2	0.10	-2.7	0.09	-1.3	0.19	-5.8	0.21	-5.6
Carphone (QCIF)	0.04	-1.0	0.66	-18.4	0.07	-2.1	0.79	-22.3	0.84	-20.7
Silent (CIF)	0.02	-1.0	0.35	-15.4	0.07	-2.4	0.40	-17.3	0.45	-19.1
Hall (CIF)	0.02	-0.5	0.32	-8.6	0.10	-2.6	0.37	-9.8	0.46	-11.2
Mobile Calendar (HD-1280x720)	0.03	-2.4	0.19	-6.8	0.06	-2.7	0.25	-9.3	0.26	-10.1
Average	0.03	-0.96	0.33	-11.5	0.09	-3.19	0.40	-13.5	0.44	-14.8

4.1 Experiments with 4x4 Intra Modes Only

In this experiment all frames are intra coded and only 4x4 mode is enabled. The performance comparisons are presented in Table 5. In these tables, a positive value indicates increment and a negative value represents decrement. As shown in Table 5(a), the proposed IDCP improves 0.09 dB PSNR and reduces bit rate by 3.19% whereas DWP improves PSNR by only 0.03 dB and reduces bit rate by only 0.96% of the original encoder. In terms of computation, the proposed IDCP increases the encoding and decoding time by 4.01% and 2.51%, respectively. In case of SMP method, the average PSNR improvement is about 0.33 dB and average bit rate reduction is about 11.5%. Whereas in our proposed ANM method, the average PSNR improvement is about 0.40 dB and average bit rate reduction is about 13.5%. The proposed ANM only method also reduces the computation of the original encoder by 41.5%. Although this method introduces some extra computations of the decoder side, the simulation results

Table 5(b). Complexity comparisons of proposed methods (only 4x4 modes, All I frames)

	DWP[4]		SMP [6]		Proposed IDCP only		Prop ANM only		ANM+IDCP	
	ΔT_1 %	ΔT_2 %	ΔT_1 %	ΔT_2 %	ΔT_1 %	ΔT_2 %	ΔT_1 %	ΔT_2 %	ΔT_1 %	ΔT_2 %
Grand Mother (QCIF)	2.22	2.25	-39.7	2.09	4.35	4.39	-52.1	1.99	-44.9	4.7
Sales man (QCIF)	1.17	0.04	-31.2	1.19	3.70	1.12	-37.9	0.91	-34.7	2.3
Stefan (QCIF)	1.01	0.39	-17.9	0.39	3.46	0.54	-25.4	0.33	-21.0	1.1
Carphone (QCIF)	1.29	1.18	-33.8	0.42	4.27	2.65	-46.0	0.39	-43.5	3.2
Silent (CIF)	1.34	0.99	-35.8	1.74	4.77	2.81	-45.8	1.53	-41.6	3.1
Hall (CIF)	1.35	0.35	-38.8	1.45	3.16	1.71	-48.5	1.18	-46.1	3.2
Mobile Calendar (HD-1280x720)	2.62	1.28	-27.6	1.55	4.39	1.84	-34.7	1.59	-32.9	2.8
Average	1.57	0.93	-32.2	1.27	4.01	2.51	-41.5	1.13	-37.8	2.91

of Table 5(b) confirm that, the computational overhead of the decoder is very low (about 1.13%). It is shown that if we combine our both methods together, about 14.8% bit rate reduction is achieved along with a 0.44 dB improvement in PSNR in the expense of 2.91% increment of decoding time. The proposed method reduces 37.8% computation of encoder.

4.2 Experiments with All Intra Modes

In this experiment all frames are encoded by intra coding and all intra modes (4x4, 8x8, and 16x16) are enabled. The results are tabulated in Table 6. Here proposed IDCP method is applied in 4x4 block and ANM 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 0.37 dB PSNR and 12.2% bit rate saving, with a maximum for sequence *Carphone* with 0.79 dB and 17.7%. We have seen that the proposed method reduces 31.2% computation of original encoder. The computation increment of decoder side is very low and that is 2.3% on average.

Table 6. Experimental results of proposed methods (All I frames, all Intra modes)

Sequence	Δ P in dB	ΔR %	ΔT_1 %	ΔT_2 %
Grand Mother (QCIF)	0.33	-14.2	-36.8	4.0
Sales man (QCIF)	0.40	-13.1	-29.0	1.4
Stefan (QCIF)	0.18	-5.8	-17.9	1.1
Carphone (QCIF)	0.79	-17.7	-33.8	3.8
Silent (CIF)	0.36	-12.3	-37.1	3.1
Hall (CIF)	0.31	-9.1	-34.7	1.3
Mobile Calendar (HD-1280x720)	0.20	-13.1	-29.2	2.0
Average	0.37	-12.2	-31.2	2.3

5 Conclusions

In this paper, we propose two methods to improve the RD performance of H.264/AVC intra encoder. At first, a distance based improved DC prediction is utilized to better representation of smooth region of sequences. Then a bit rate reduction scheme for representing the intra prediction mode is described. The proposed methods not only improve the RD performance but also reduce 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. Shiodera, T., Tanizawa, A., Chujoh, T.: Block based extra/inter-polating prediction for intra coding. In: Proc. IEEE ICIP 2007, pp. VI-445–VI-448 (2007)

3. Ye, Y., Karczewicz, M.: 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 (2008)
4. Yu, S., Gao, Y., Chen, J., Zhou, J.: Distance based weighted prediction for H.264 Intra Coding. In: Proc. ICALIP 2008, pp. 1477–1480 (2008)
5. Wang, L., Po, L.M., Uddin, Y.M.S., Wong, K.M., Li, S.: A novel weighted cross prediction for H.264 intra coding. In: Proc. IEEE ICME 2009, pp. 165–168 (2009)
6. Kim, D.Y., Han, K.H., Lee, Y.L.: Adaptive Single-Multiple Prediction for H.264/AVC Intra Coding. IEEE Trans. on Circuit and System for Video Tech. 20(4), 610–615 (2010)
7. JM reference software 12.4, `http://iphome.hhi.de/suehring/tml/download/old_jm/jm12.4.zip`
8. Bjontegaard, G.: Calculation of average PSNR differences between RD-curves. Presented at the 13th VCEG-M33 Meeting, Austin, TX (April 2001)
9. Sarwer, M.G., Po, L.M., Wu, J.: Fast Sum of Absolute Transformed Difference based 4x4 Intra Mode Decision of H.264/AVC Video Coding Standard. Journal of Signal Processing: Image Commun. 23(8), 571–580 (2008)
10. Sarwer, M.G., Po, L.M., Wu, J.: Complexity Reduced Mode Selection of H.264/AVC Intra Coding. In: Proceeding on International Conference on Audio, Language and Image Processing (ICALIP 2008), China, pp. 1492–1496 (2008)
11. Sarwer, M.G., Wu, Q.M.J.: Adaptive Variable Block-Size Early motion estimation termination algorithm for H.264/AVC Video Coding Standard. IEEE Trans. Circuit and System for Video Technol. 19(8), 1196–1201 (2009)
12. La, B., Jeong, J., Choe, Y.: Most probable mode-based fast 4×4 intra-prediction in H.264/AVC. In: International Conference on Signal Processing and Communication Systems, ICSPCS 2008, pp. 1–4 (2008)
13. Elyousfi, A., Tamtaoui, A., Bouyakhf, E.: A New Fast Intra Prediction Mode Decision Algorithm for H.264/AVC Encoders. Journal of World Academy of Science, Engineering and Technology 27, 1–7 (2007)
14. La, B., Jeong, J., Choe, Y.: Fast 4×4 intra-prediction based on the most probable mode in H.264/AVC. IEICE Electron. Express 5(19), 782–788 (2008)
15. Krishnan, N., Selva Kumar, R.K., Vijayalakshmi, P., Arulmozhi, K.: Adaptive Single Pixel Based Lossless Intra Coding for H.264 / MPEG-4 AVC. In: International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), vol. 3, pp. 63–67 (2007)