Fast Intra Prediction in the Transform Domain

Chanyul Kim and Noel E. O'Connor

CLARITY:Centre for Sensor Web Technologies, Dublin City University, Glasnevin, Dublin, Ireland

Chanyul.kim,oconnorn@eeng.dcu.ie

Abstract

In this paper, we present a fast intra prediction method based on separating the transformed coefficients. The prediction block can be obtained from the transformed and quantized neighboring block generating minimum distortion for each DC and AC coefficients independently. Two prediction methods are proposed, one is full block search prediction (FBSP) and the other is edge based distance prediction (EBDP), that find the best matched transformed coefficients on additional neighboring blocks. Experimental results show that the use of transform coefficients greatly enhances the efficiency of intra prediction whilst keeping complexity low compared to H.264/AVC.

I. INTRODUCTION

Intra prediction is commonly used to improve the coding efficiency of intra coding in video codecs. It utilizes the spatial correlation in an image to predict the block being encoded from its surrounding pixels. Spatial domain intra prediction was first introduced in H.264/AVC [1]. To achieve the highest coding efficiency possible, various methods of prediction, corresponding to different prediction directions, have been adopted in H.264/AVC. It is recognized to be one of the main factors contributing to the success of H.264/AVC. A Rate Distortion Optimization (RDO) technique is used in H.264/AVC for selecting the actual prediction mode used of those available, meaning that the encoder must search all the possible prediction modes exhaustively in order to encode blocks. As a result, the computational complexity in H.264/AVC is dramatically increased. The prediction pixels are predicted from surrounding pixels by low-pass filtering, termed extrapolation in H.264/AVC. However, it is not sufficient to represent prediction blocks only using extrapolation when the content inside a block is not homogenous or features complex texture. Much research has been thus performed into two major interesting topics in intra coding. One is the design of fast prediction algorithms; the other is enhancing existing algorithms towards greater coding gain even at the cost of increasing computational complexity.

A. Fast Algorithms

A number of efforts have been made to develop fast intra prediction algorithms for H.264/AVC based on RDO. However, these methods pose real challenges for implementing real-time video compression due to their complexity. The Joint Model reference software[1] (JM) provides a hint for reducing computational complexity by not using RDO but rather using a configuration entitled Low-Complexity-Mode (RDO off). However, this clearly suffers from a substantial decrease in coding gain, even if it lowers complexity by a factor of 10, as illustrated later in this paper (Table. II). However, even with this decrease in complexity, the result is still far from requirements for some fast encoding and real-time applications. The most common algorithm used in fast intra prediction is to eliminate candidate modes using side information such as the edge directional information contained in the block. Pan *et al.* [2] proposed a fast intra mode decision scheme. They calculated an edge direction histogram using the Sobel operator to predict the primary prediction mode. Only a small number of intra prediction modes satisfying the primary prediction mode are used thereby reducing the total number of mode decisions necessary. Wang *et al.* [3] proposed a fast intra mode decision algorithm based on DCT edge characteristics. They calculated the angle of the edge in the block using seven DCT coefficients in order to choose the best modes. Tian *et al.* [4] also suggested

---

[1]The source code is available at $http://iphome.hhi.de/suehring/tml/download/old\_jm/$

a fast algorithm based on the Prewitt operator where they extracted two dominant edge directions in each block.

These methods all have a common approach to reduce computational complexity, that is, edge directional features are extracted by various methods before mode decisions and are used to eliminate candidate modes. The average reduction ratio of these approaches is about 70%, and this is an enormous contribution to fast encoding. However, they still require a pre-processing step in order to detect directional features. So, Jun Sung *et al.* [5] proposed a fast algorithm using 16×16 prediction modes for obtaining 4×4 predictions. In their approach, the directional information is estimated in a bigger block (16×16) and applied to small blocks (4×4). Taeyoung *et al.* [6] suggested a fast algorithm to find the best mode by updating a context adaptive conditional probability table indicating the most probable modes of the current frame, and updating this table for every frame. Their approaches did not use a pre-processing step compared to [2], [3], [4]. In spite of these efforts, it is still hard to implement a real-time H.264/AVC encoder, even at CIF (352×288) resolution, without hardware and algorithm optimizations. Therefore, computationally efficient tools for exploiting spatial correlation will be the one of main requirements in next generation video codecs.

B. Algorithms for Improved Coding Efficiency

The intra prediction approach is to copy neighboring pixel values into the current block along a specified direction, followed by simple low-pass filtering using adjacent pixels. Such a simple prediction method is not sufficient to truly exploit spatial correlation. Tan *et al.* [7] presented intra prediction based on texture synthesis via template matching in a region of reconstructed pixels. They also presented an enhanced version of intra prediction via directional template matching and averaging [8]. Their approach showed an improvement in coding gain of 15% compared to H.264/AVC intra coding, but the computational complexity increased dramatically since the template matching algorithm is similar to a motion search algorithm in inter frame coding. Recently, Liu *et al.* [9] presented an intra prediction algorithm based on edge inpainting. Inpainting is a restoration technique in computer vision and graphics that requires significant computation as it needs pre-processing such as solving partial differential equations and texture synthesis.

Apart from coding efficiency, low complexity attributes of compression such as low memory utilization and power consumption are increasingly important in commercial applications. However, the challenge comes in finding the optimum trade-offs as the requirements are not comfortable bed-fellows. The method proposed in this paper focuses on fast intra prediction whilst maintaining reasonable coding efficiency. Compared to intra prediction in H.264/AVC, we utilize separated transformed coefficients instead of the directional modes. Two intra prediction methods named Full Block Searching Prediction (FBSP) and Edge Block Distance Prediction (EBDP) are presented. FBSP uses AC and DC prediction by dividing a 4×4 sub-block into separated low (DC) and high frequency (AC) transformed coefficients, and finding the neighboring block that generates the minimum Sum of Absolute Transformed Differences (SATD). EBDP is introduced to reduce complexity for performing SATD operations with all neighboring blocks. The rest of this paper is organized as follows. Section II presents the process of H.264/AVC intra prediction, the proposed intra prediction is described in Section III. Extensive experimental results are presented in Section IV to compare the performance of the proposed method with that of H.264/AVC in terms of PSNR and complexity.

II. INTRA VIDEO CODING ARCHITECTURE IN H.264/AVC

Intra coding refers to the case where only spatial redundancies of neighboring Macro-Blocks (MBs) within a video picture are exploited. In H.264/AVC intra coding, two intra MB modes for luma are
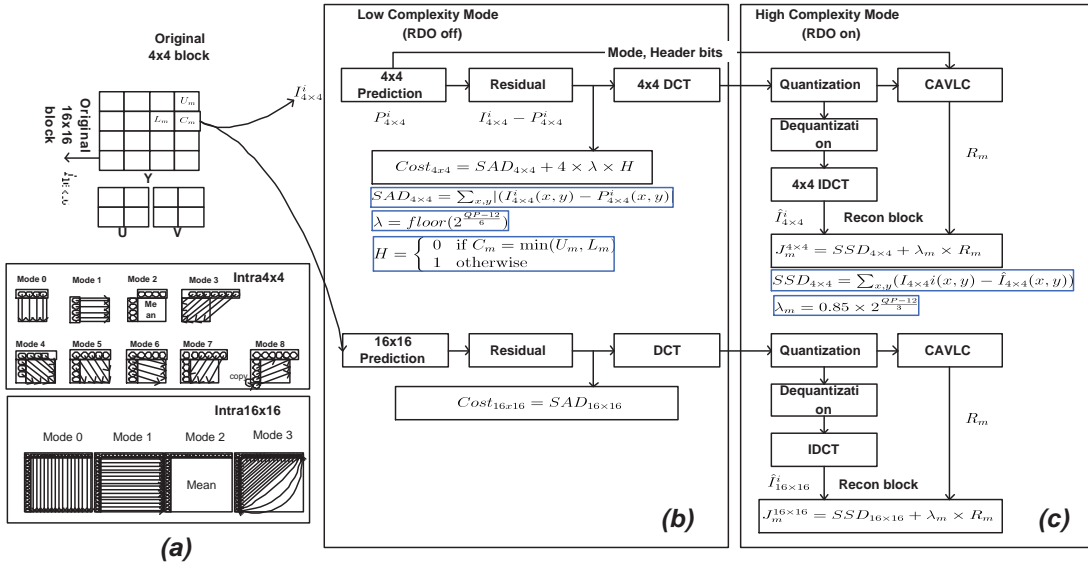
Fig. 1. Graphical functional flow of H.264/AVC Intra Coding; (b) The SAD (sum of Absolute Difference) between original and predicted block is used for measuring cost function in Low-Complexity-Mode. (c) SSD(sum of the squared differences, denoted in Eq. (1)) is used in High-Complexity-Mode. (a) There are 9 modes for 4x4 prediction, 3 modes for 16x16 prediction; i indicates the ith 4x4 block in MB. P, I and $\hat{I}$ represent prediction, original and reconstructed block.

supported in each profile. One is Intra4x4 prediction mode, and the other is Intra16xl6 prediction mode depicted in Fig. 1(a). Intra8x8 is a new intra prediction type defined in H.264/AVC FRExt [10]. For Intra4x4, the MB is divided into 16 non-overlapping 4×4 luma blocks and each 4×4 block can select one of nine prediction modes. For Intra16xl6, each MB can select one of the four modes. Chroma Intra prediction is independent of luma. Two chroma components are simultaneously predicted by one mode only. The possible chroma prediction modes are very similar to those of Intra16x16 except for different block size (8×8) and the index of DC mode. The Intra4x4 mode can predict a block more accurately but requires more bits to represent the mode information than Intra16xl6. So, Intra4x4 tends to be used for highly textured regions while Intra16xl6 tends to be chosen for plain regions. Mode decision is not specified in the H.264/AVC standard, but is arguably the most important step at the encoder side because of its affect on coding efficiency. On the contrary, it consumes lots of processing time and memory accesses. Fortunately, H.264/AVC reference software (JM) can reduce computational complexity dramatically by forbidding RDO and the Hadamard transform in the coding process, which means that the parameters *RDOptimization* and *UseHadamard* in JM are set to zero. This parameter configuration is called Low-Complexity-Mode, whilst RDO is also called High-Complexity-Mode. The difference between these two is which cost functions are used for selecting the mode. In Low-Complexity-Mode, a Sum of Absolute Difference (SAD) based cost function is used to estimate the final distortion. The detailed procedures in Low-Complexity-Mode are outlined below and graphical flows are depicted in Fig. 1(b).

1) Generate a 4×4 prediction block($P_{4 \times 4}^i$) using encoded neighboring pixels followed by extrapolation in all modes $m \in [0, 8]$.
2) Calculate SAD ($SAD_{4 \times 4}$) between the original 4×4 block($I_{4 \times 4}^i$) and the predicted block ($P_{4 \times 4}^i$).
3) Compute $Cost_{4 \times 4} = SAD_{4 \times 4} + 4 \times \lambda \times H$, where λ is an exponential function of the quantization factor, H equals 0 for the most probable mode ($C_m = min(U_m, L_m)$), elsewhere 1.
4) Repeat 1) to 3) for all 9 modes, and choose the one that has the minimum cost.
5) Calculate $CostMB_{4 \times 4} = \sum_{i=1}^{16} Cost_{4 \times 4}^i + 24\lambda$ for a MB.
6) Generate a 16×16 prediction block ($m \in [0, 3]$).
7) For the residues, perform Hadamard transform for each 4×4 block. This is an optional procedure.

8) Repeat 6) to 7) for all 4 modes and choose the one with minimum cost as the $CostMB_{16\times16}$.

9) If $CostMB_{16\times16} \geq CostMB_{4\times4}$ is true, the 4×4 prediction mode will be selected for the current MB, otherwise, 16×16 prediction mode is selected.

The cost function and SAD operations should be replaced by $J_m^{\min} = \min(SSD_m + \lambda_m \times R_m)$ at mode m and SSD in the High-Complexity-Mode (RDO on) shown in Fig. 1(c), where SSD can be obtained using Eq (1).

$$SSD = \frac{1}{MN} \sum_{i=0}^{i=M} \sum_{j=0}^{j=N} \left(I(i,j) - \widehat{I(i,j)} \right)^2 \tag{1}$$

where $I(i,j)$ and $\widehat{I(i,j)}$ are a original and reconstructed pixel at (i,j) respectively, and M, N represent the maximum number of row and column. The decision procedure is the same as that of Low-Complexity-Mode.

From a complexity point of view, H.264/AVC encodes MBs by iterating all the luma intra decisions for each possible chroma intra prediction mode for the best coding efficiency. Therefore, the number of mode combinations for luma and chroma components in a MB is $C8 \times (L4 \times 16 + L16)$, where $C8, L4$, and $L16$ represent the number of modes for chroma prediction, 4×4 luma prediction and 16×16 prediction respectively. This means that, $4 \times (9 \times 16 + 4) = 592$ different RDO calculations have to be performed before a best RDO mode is determined. If the 8×8 luma prediction of H.264/AVC FRExt is included, the number of mode combinations is $C8 \times (L4 \times 16 + L8 \times 4 + L16) = 4 \times (9 \times 16 + 9 \times 4 + 4) = 736$. Even in Low-Complexity-Mode, $9 \times 16 + 4 + 4 \times 2 = 156$ mode decisions are needed. Although Low-Complexity-Mode can greatly reduce computational load, encoding time still needs to be reduced for some applications requiring very low delay.

III. THE PROPOSED INTRA PREDICTION AND CODING TECHNIQUE

The proposed fast intra prediction uses transformed data from neighboring blocks, and obtains fast coding with reasonable efficiency compared to H.264/AVC. We present two prediction methods, and analyze the intra prediction error caused by imperfect prediction and drift propagation as mentioned in [11].
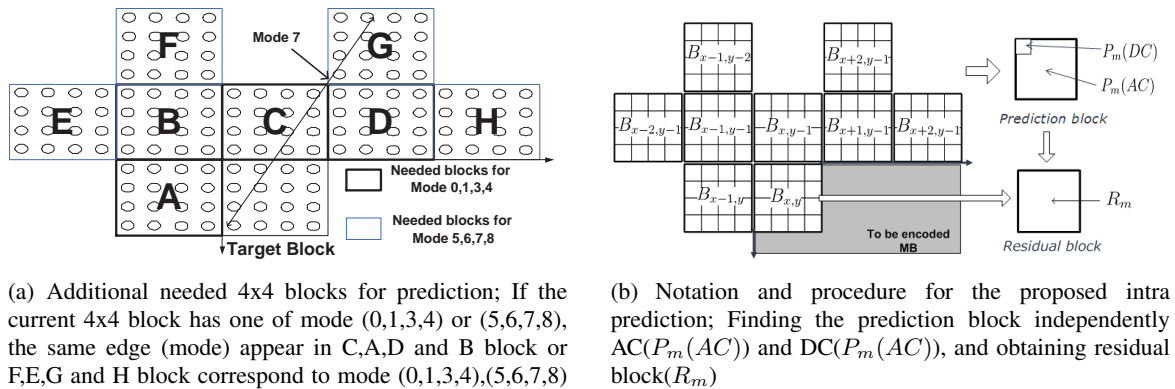
A. Motivation

(a) Additional needed 4x4 blocks for prediction; If the current 4x4 block has one of mode (0,1,3,4) or (5,6,7,8), the same edge (mode) appear in C,A,D and B block or F,E,G and H block correspond to mode (0,1,3,4),(5,6,7,8)

(b) Notation and procedure for the proposed intra prediction; Finding the prediction block independently $AC(P_m(AC))$ and $DC(P_m(AC))$, and obtaining residual block(R_m)

Fig. 2. Neighboring blocks

The underlying assumption is that edge direction is not severely changed within two 4×4 blocks. When the current 4×4 block has one of mode (0,1,3 and 4) as defined in H.264/AVC, the same edges

potentially appear in neighboring blocks (C,A,D and B). Extra neighboring blocks (F,E,G and H) are needed for other modes (5,6,7 and 8). For example, if the current block has mode 7, the block containing the same edge appears in block G depicted in Fig. 2(a). G could be a good sampler for predicting a block if it has a similar edge direction compared to the current block.
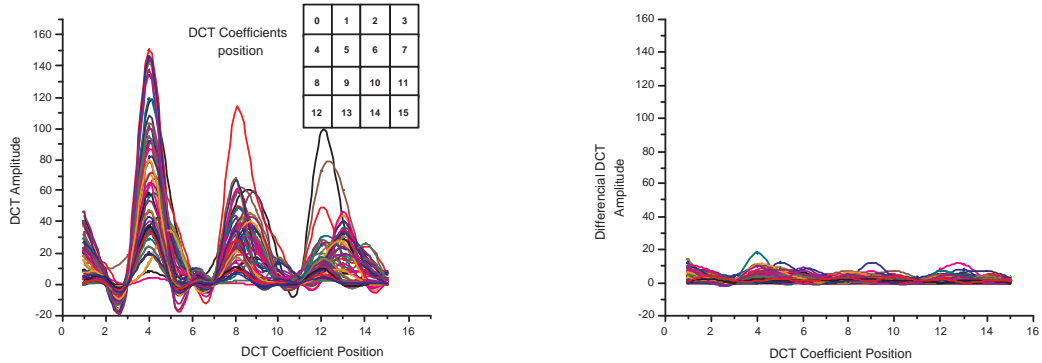
(a) Distribution of AC DCT coefficients of the current block($B_{x,y}$ in Fig. 2(b)) for Mode 1 in the first frame of the Foreman sequence

(b) Residual DCT coefficients are decreased; the differential distribution of DCT coefficients between the current block and predicted block (($B_{x,y} - B_{x-1,y}$) if $B_{x-1,y}$ has mode 1)

Fig. 3. Distribution of DCT coefficients (a) in the current block (mode 1) (b) after using an appropriate neighbor block (mode 1) as a predictor

Fig. 3 shows the distribution of DCT coefficients in a block which has mode 1 (horizontal direction in the 4×4 prediction mode of H.264/AVC) in the first frame of the Foreman sequence and its residual from a prediction block that has the same mode. The distribution of DCT coefficients has a relative maximum value at position 4,8 and 12 as shown in Fig 3(a). When a neighboring block has been chosen for the same mode 1, and subtracted, the distribution of DCT residual coefficients decrease dramatically as depicted in Fig 3(b). Clearly, selecting this block as a predictor reduces the entropy of the data to be encoded.

B. The Proposed Intra Prediction Scheme

Two intra prediction methods are proposed in this paper; FBSP and EBDP. EBDP is targeted for reducing SATD computation time in the FBSP by estimating edge direction with DCT coefficients.

Full Block Search Prediction (FBSP)

FBSP finds the block that generates minimum distortion by separating AC and DC coefficients of neighboring blocks. The selective mode indicates not the directional feature mentioned in H.264/AVC but rather the location of the predicted block outlined in Table I. Illustrated in Fig. 2(b), let $B_{x,y}$ be the current transformed and quantized 4×4 block to be encoded. First of all, DC prediction ($P_m(DC)$) and AC prediction ($P_m(AC)$) are calculated to find the index of the block satisfying minimum distortion (SATD) between the current block and neighboring blocks as specified in Eq (2).

$$P_m(AC) = \arg\min_{\in R}\left[\sum_{k=1}^{k=15}|B_{x,y}(k) - B_{x-i,y-j}(k)|\right]$$

$$P_m(DC) = \arg\min_{\in R}\left[|B_{x,y}(0) - B_{x-i,y-j}(0)|\right]$$

$$R : (i, j, m) = \{(1,0,0),(0,1,1),(2,1,2),(1,1,3),$$
$$(-1,1,4),(-2,1,5),(-1,-2,6),(1,2,7)\}$$

(2)

Mode	Block Notation	Block Position
0	$B_{x,y-1}$	Upper
1	$B_{x-1,y}$	Left
2	$B_{x-1,y-1}$	Upper-Left
3	$B_{x+1,y-1}$	Upper-Right
4	$B_{x-2,y-1}$	Upper-two-Left
5	$B_{x-1,y-2}$	two-Upper-Left
6	$B_{x+2,y-1}$	Upper-two-Right
7	$B_{x+1,y-2}$	two-Upper-Right

TABLE I

PREDICTION MODE ALLOCATION

where $P_m(AC), P_m(DC)$ are AC and DC prediction at mode m, $B_{x,y}(k)$ is the kth DCT coefficients of the 4×4 block and R is a total set of neighboring blocks' position (modes). After determining the prediction mode, residual DCT coefficients can be calculated as in Eq (3), and are sent to the entropy coder (CAVLC and CABAC).

$$R_m(0) = (B_{x,y}(0) - P_m(DC)) \quad \textbf{if } k = 0$$
$$R_m(k) = (B_{x,y}(k) - P_m(AC)_k) \quad \textbf{if } k = [1, 15]$$

(3)

where $R_m(k)$ is residual k^{th} DCT coefficient at mode m.

Edge Based Distance Prediction (EBDP)

It is well known that the horizontal and vertical edge features of a block, including orientation, can be estimated from the DCT coefficients as shown in Fig. 4(a) [12], [13]. EBDP finds the neighboring block (one of eight blocks) that generates minimum edge distance (difference) compared to the current block. We define the parameter D_{tan} D_{cot} as the tangential and cotangential direction of the edge and σ_b as the standard deviation of AC DCT coefficients as follows.

$$D_{tan} \overset{\text{def}}{=} \left[2 \times \frac{\sum_{k=1}^{3} |C_k|}{\sum_{k=1}^{3} |C_{4k}|} \right] \qquad D_{cot} \overset{\text{def}}{=} \left[2 \times \frac{\sum_{k=1}^{3} |C_{4k}|}{\sum_{k=1}^{3} |C_k|} \right], \quad \text{if } (|D_{tan}| > 2) \bigcup (\sum_{k=1}^{3} |C_{4k}| = 0)$$

$$\sigma_b^2 \overset{\text{def}}{=} \frac{\sum_{k=1}^{15} C_k^2}{15} - \left(\frac{\sum_{k=1}^{15} C_k}{15} \right)^2$$

(4)

where C_k is the kth DCT coefficient depicted in Fig. 4(a), $[.]$ is greater integer and $\bigcup$ is the Logical OR operation. A circle divided in 16 segments ($\frac{2\pi}{16} = \frac{\pi}{8}$) is the reference unit for counting edge distance. Edge distance can then be measured by defining direction D using D_{tan}, D_{cot} as shown in Fig. 4(b).

The algorithm for Edge Based Distance Prediction as follows

1) Find D_{tan} and σ_b, if $|D_{tan}| > 2 \bigcup (\sum_{k=1}^{3} |C_{4k}| = 0)$, D_{cot} is calculated.
2) If σ_b is smaller than a pre-defined threshold (τ), a block is considered as a non-edge or a plain region, D is set to a sufficiently big number (we set D to 10 in our method), otherwise D is allocated a number (0 to 7) according to D_{tan}, D_{cot} depicted in Fig. 4(b).
3) Find the minimum edge distance comparing to each neighboring block. Fig. 4(c) shows an example for edge distance measurement. The edge distance could be obtained by finding $\min(D*, 8 - D*)$.
4) Obtain AC prediction block to find the index to satisfy;

$$P_m(AC) = \begin{cases} \arg\min_{\in R} |\min(D*, 8 - D*)| & \text{if } D_{cur} \neq 10 \\ \arg\min_{\in R} |D_{cur} - D_{nei}| & \text{elsewhere} \end{cases}$$
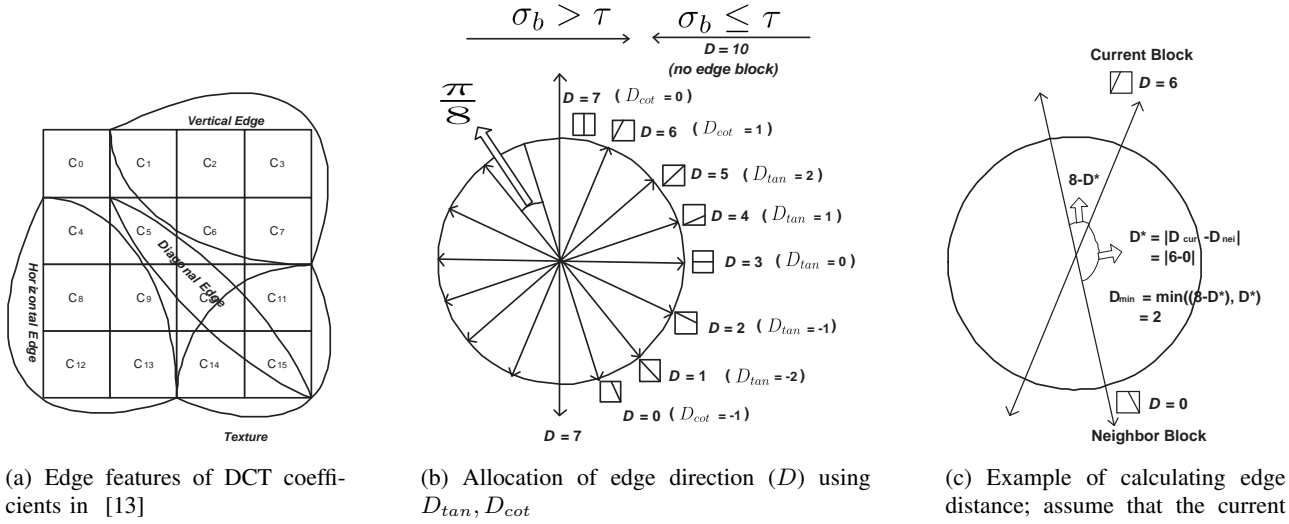
(a) Edge features of DCT coefficients in [13]

(b) Allocation of edge direction (D) using D_{tan}, D_{cot}

(c) Example of calculating edge distance; assume that the current block direction has $D = 6$, the neighbor block has $D = 0$, the difference($D*$) is 6 and the edge distance (D_{min}) is 2

Fig. 4. DCT feature, direction allocation and an example for calculating edge distance.

$D*$ indicates the edge difference, D_{cur} and D_{nei} represent the edge direction (D) of the current and the neighbor block as shown in Fig. 4(c). R is denoted in Eq. (2).

5) Find DC prediction and residual using the same method in FBSP.

C. Intra Prediction Error

Let I_n denote the nth MB to be coded in a video and P_n denote the corresponding intra prediction signal. The reconstructed MB in H.264/AVC intra coding can be obtained as in Eq. (5), and is affected not only by the quantization error factor (σ_q) but also the accuracy of the prediction block (P_n).

$$
\begin{aligned}
\hat{I}_n &= T^{-1}\left(Q^{-1}\left(Q\left(T(I_n - P_n)\right)\right)\right) + P_n \\
&\cong (I_n - P_n) \times \sigma_q + P_n \\
&= I_n \times \sigma_q + (1 - \sigma_q) \times P_n
\end{aligned}
\tag{5}
$$

where $T(), T^{-1}(), Q(), Q^{-1}()$ represents the function for the discrete cosine transform (DCT), inverse discrete cosine transform (IDCT), quantization and inverse quantization, respectively, $\sigma_q \in [0, 1]$ is quantization error factor noted as $Q^{-1}(Q(.))$. $T^{-1}(T(.))$ can be approximated as one due to the exact match integer transform in H.264/AVC.

On the contrary, the reconstructed MB using the proposed intra prediction can be estimated by

$$
\begin{aligned}
\hat{I}_n &= T^{-1}\left(Q^{-1}\left(Q\left(T(I_n - P_n)\right) + Q\left(T(P_n)\right)\right)\right) \\
&= T^{-1}\left(Q^{-1}\left(Q\left(T(I_n)\right)\right)\right) \\
&\cong I_n \times \sigma_q
\end{aligned}
\tag{6}
$$

The reconstructed MB is affected only by quantization error factor regardless of any type of prediction block. Therefore, the proposed intra prediction shows small SSD error for all values of QP as illustrated in Fig. 5(a). The SSD error is quite small in both cases at low QP, but the SSD error in H.264/AVC increases dramatically for high QP since it affects the quantization error factor and the prediction block as
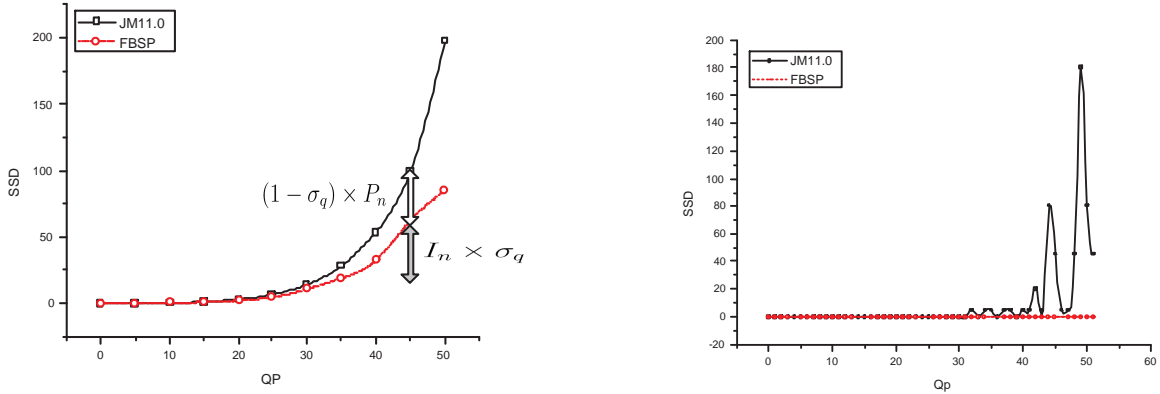
(a) SSD error vs QP in the 1st frame of Foreman; FBSP shows a small SSD error compared to H.264/AVC for all QP, with especially good SSD error for high QPs

(b) Drift propagation for a uniform frame (all luminance Y are set to 100); the SSD error has a damping effect over specific QP values in JM

Fig. 5. The characteristic response of QP in JM with FBSP

shown in Eq. (5). This means that the proposed intra prediction shows good performance in low bit-rate compression as shown in Fig. 6 and Fig. 7.

Rounding of predicted pixels may introduce distortion since it involves sacrificing part of the signal fidelity. When the intra prediction starts from position (0,0), only DC prediction is available. This generates errors $((1 - \sigma_q) \times P_n)$, and affects the next block. This is termed drift propagation. Since a plain region is more affected by drift, in order to check the effect of drift propagation, a sequence is generated with all Y,U,V pixels set to 100 and encoded using the JM reference software. Drift propagation is one of the factors that may result in an increase in distortion even though QP is decreased as illustrated in damping the SSD error of Fig 5(b). The proposed intra prediction has no drift propagation due to its independence of the predicted signal as depicted in Eq (6).

IV. IMPLEMENTATION AND EXPERIMENTAL RESULT

To evaluate the performance of the proposed method, our approach was implemented in the H.264/AVC reference software JM11.0 with four different test sequences (Foreman, Hallmonitor, Mother and Daughter and Mobile at CIF resolution). All experiments are performed with 100 frames in both Low-Complexity and High-Complexity modes using only I frames. The proposed fast intra prediction is written in ANSI C++ and Intel Integrated Performance Primitive 5.3 library [14]. All tests are performed on a Intel Core(TM)2 Duo 3.0GHz with 2GB RAM based on Window XP version 2002 with service pack 2.

A. Comparison of RD Performance

In order to investigate the effect of the RD optimization routine in the JM reference software, sequences are tested with the rate-distortion function on and off. FBSP and EBDP mentioned in Section III-B are compared to the performance of H.264/AVC.

Fig. 6 shows the RD performances for Foreman and HallMonitor. The proposed methods show degradation of not less than 0.8dB at 600kbps, but shows improved performance at low bit-rates (250kbps for Foreman, 350kbps for Hall Monitor) since JM suffers the degradation from not only QP and its predicted signal as shown in Eq. (5). Our proposed method uses 16 modes (8 modes for DC, 8 modes for AC) including DC and AC prediction, compared to 13 modes (9 modes for 4×4, 4 modes for 16×16) in H.264/AVC. It increases the required bits (R) and keeps almost the same SSD for low QP as shown in
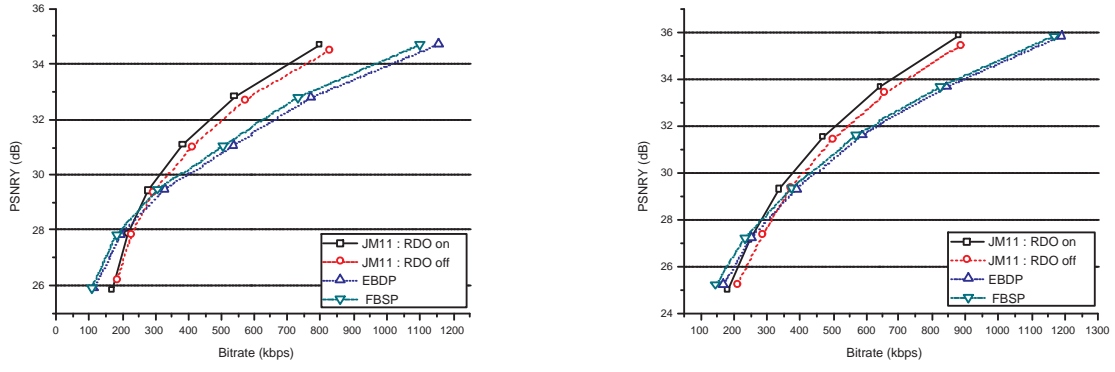
(a) Foreman@352×288; FBSP shows PSNR degradation of 0.8dB compared to JM @600kbps, 0dB@250kbps

(b) HallMonitor@352×288; PSNR degradation of 0.5dB compared to JM @600kbps, 0dB@350kbps

Fig. 6. R-D curve; FBSP has a degradation compared to JM in most bit-rate regions
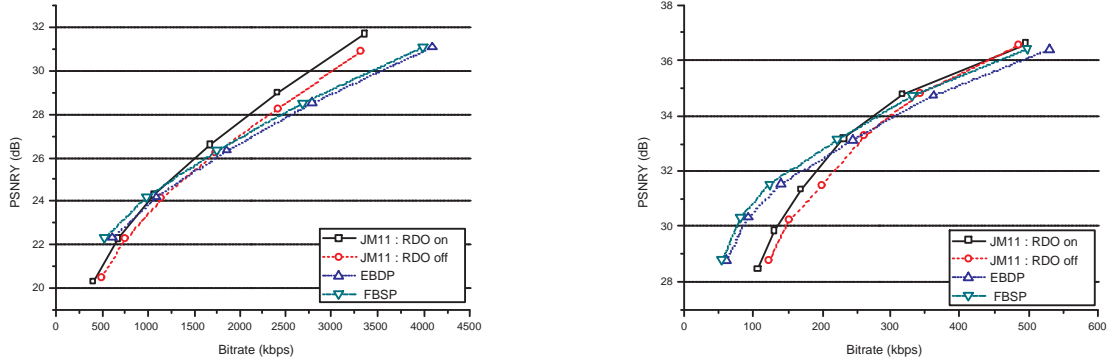
(a) MobileCalendar@352×288; FBSP dose not show much PSNR degradation in all bit-rate regions

(b) MotherDaughter@352×288; FBSP show better performance; especially an almost 2dB gain @125kbps

Fig. 7. R-D curve: The proposed method shows the same or better quality compared to JM

Fig. 5(a). The RD performance of the proposed method is poor for low QP and good for high QP for changing Lagrange minimum cost;

$$J_{min} = SSD + \lambda \times R \tag{7}$$

The proposed method shows the same or nearly as good performance for the Mobile and Mother sequences as shown in Fig. 7. The mobile sequence is a complex scene in terms of edges, the prediction error($(1 - \sigma_q)P_n$) could be more dominant rather than caused by quantization (σ_q), so the proposed method shows the same or better performance. The Mother and daughter sequence has a lot of plain regions which causes drift propagation at high QP, the proposed method shows a 2dB PSNR performance at 125kbps due to its immunity to error characteristics as shown in Fig. 7(b).

B. Profiling Execution Time

Table II shows the result of profiling the proposed methods and JM. In the JM, rate distortion optimization (RDO) is the most computationally demanding block. Without RDO, the execution time increases almost eight times compared to high complexity mode. The FBSP approach increases the frames per

second by a factor of five compared to H.264/AVC with low complexity mode due to its independency of extrapolation and mode decisions.

V. Conclusion

A fast intra prediction scheme based on separating the transformed coefficients is presented in this paper. Neighboring blocks are selected based on comparing DCT coefficients. The approach leverages the estimated edge direction in the block and as a result requires consideration of extra neighboring blocks. The intra prediction error is analyzed in term of distortion (SSD) and drift propagation. The experimental results confirm that the proposed method provides a valid way for improving the design of intra prediction. It has been shown that the use of transform coefficients greatly enhances the efficiency of intra prediction whilst keeping complexity low.

Unit (fps:frame per second)	JM11.0 Reference		The proposed method	
Sequences	High Complexity Mode	Low Complexity Mode	FBSP	EBDP
Foreman	1.44	12.47	74.5	93.2
Hall Monitor	1.43	12.15	73.2	95.1
Mobile	1.14	11.49	64.3	86.3
Mother and Daughter	1.52	11.93	75.8	96.4

TABLE II

AVERAGE EXECUTION TIME FOR 30 FRAMES WHICH ARE ALL INTRA PICTURE : UNIT FRAME PER SECOND

REFERENCES

[1] G. Sullivan, P. Topiwala, and A. Luthra, "The h.264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions," SPIE Conference on Applications of Digital Image Processing XXVII, Special Session on Advances in the New Emerging Standard: H.264/AVC, Tech. Rep., 2004.

[2] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, and S. Wu, "Fast mode decision algorithm for intraprediction in h.264/AVC video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 7, pp. 813–822, July 2005.

[3] Z. Wang, J. Yang, Q. Peng, Z. Ma, and C. Zhu, "A fast transform domain based algorithm for h.264/AVC intra prediction," in *Multimedia and Expo, 2007 IEEE International Conference on*, Beijing, July 2007, pp. 1563–1566.

[4] G. Tian, T. Zhang, X. Wei, T. Ikenaga, and S. Goto, "An efficient fast mode decision algorithm for h.264/AVC intra prediction," in *Image and Signal Processing, 2008. CISP '08. Congress on*, vol. 1, Sanya, Hainan, May 2008, pp. 411–415.

[5] J. S. Park and H. J. Song, "Selective intra prediction mode decision for h.264/avc encoders," *WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY*, vol. 13, pp. 51–55, 2006.

[6] T. Na, M. Kim, S. Hahm, C. Park, and K. Park, "A fast 4x4 intra mode decision for inter frame coding in h.264—MPEG-4 part 10," in *Broadband Multimedia Systems and Broadcasting, 2008 IEEE International Symposium on*, Las Vegas, NV, Mar./Apr. 2008, pp. 1–5.

[7] T. K. Tan, C. S. Boon, and Y. Suzuki, "Intra prediction by template matching," in *Image Processing, 2006 IEEE International Conference on*, Atlanta, GA, Oct. 2006, pp. 1693–1696.

[8] T. Tan, C. Boon, and Y. Suzuki, "Intra prediction by averaged template matching predictors," in *Consumer Communications and Networking Conference, 2007*, Las Vegas, NV, USA,, Jan 2007, pp. 405–409.

[9] D. Liu, X. Sun, and F. Wu, "Intra prediction via edge-based inpainting," in *Data Compression Conference, 2008. DCC 2008*, Snowbird, UT, Mar. 2008, pp. 282–291.

[10] G. J. Sullivan, P. N. Topiwala, and A. Luthra, "The h.264/avc advanced video coding standard: overview and introduction to the fidelity range extensions," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, A. G. Tescher, Ed., vol. 5558, Nov. 2004, pp. 454–474.

[11] A. Leontaris and A. M. Tourapis, "Drift characterization of intra prediction and quantization in h.264," in *Data Compression Conference, 2008. DCC 2008*, Mar. 2008, pp. 212–221.

[12] K. Qiu, J. Jiang, G. Xiao, and S. Irianto, "Dct-domain image retrieval via block-edge-patterns," 2006, pp. I: 673–684.

[13] C.labit and J.P.Marescq, "Temporal adaptive vector quantization for image sequence coding," in *SPIE, Advances in Image Processing*, vol. 804, no. 371-378, Hague, Netherlands, Apr 1989.

[14] *http://www.intel.com/cd/software/products/asmo-na/eng/302910.htm*.