A FREQUENCY DOMAIN APPROACH TO INTRA MODE SELECTION IN H.264/AVC

Andy C. Yu, Graham R. Martin and Heechan Park

Department of Computer Science, University of Warwick, Coventry, CV4 7AL, United Kingdom
Email: {AndyCYu, Graham.Martin, HeeChan}@dcs.warwick.ac.uk

ABSTRACT

Multiple intra-mode prediction is one of the new features introduced in the emerging video standard, H.264/AVC. Its function is to further reduce spatial redundancies in an intra-coded macroblock prior to conventional transform coding and entropy coding. However, the process is computationally expensive especially when Lagrangian cost evaluation is employed. In this paper, we propose a fast algorithm operating in the frequency domain to accelerate intra-frame mode selection. Extensive simulations verify that the proposed algorithm outperforms several existing methods, providing speed-ups of up to 75% with insignificant degradation in picture quality.

1. INTRODUCTION

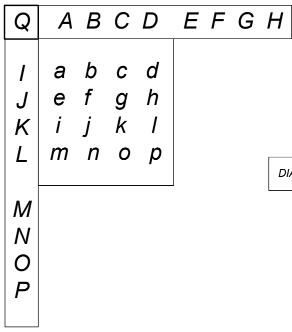
The new H.264/AVC video coding standard contains a number of advanced features. One of the new features in intra-frame coding is spatial prediction based on extrapolation from previously encoded pixels. Fig. 1 illustrates the corresponding positions of a 4x4 block containing elements *a* to *p* and its neighbouring encoded pixels, *A* to *Q*, used in the extrapolation schemes. In total nine different extrapolation schemes ('modes') are recommended [1]. Fig. 2 depicts the eight directional modes. The vertical mode, *VERT*, extrapolates a 4×4 block vertically with 4 neighbouring pixels, *A*, *B*, *C*, *D*, whereas the horizontal member, *HORT*, utilises the horizontal adjacent pixels, *I*, *J*, *K*, *L* to perform the prediction. With the exception of *DC*, the other modes operate in a similar manner, according to their corresponding orientations. The directionless member, *DC*, extrapolates all pixels as $(A+B+C+D+I+J+K+L)/8$.

The method of selecting modes in the H.264 standard requires the application of Lagrangian rate-distortion optimisation. The general equation is defined as:

$$J(\text{mode} \mid \text{Qp}) = D(\text{mode} \mid \text{Qp}) + \lambda . R(\text{mode} \mid \text{Qp}) \quad (1)$$

where J is the Lagrangian cost. D is a measure of the distortion between the original 4×4 block and the reconstructed block for each mode and R reflects the number of bits. λ represents the Lagrange parameter, which is associated with the quantisation factor, Qp, by the relationship

$$\lambda = 0.85 \times 2^{(\text{Qp}-12)/3} \quad (2)$$



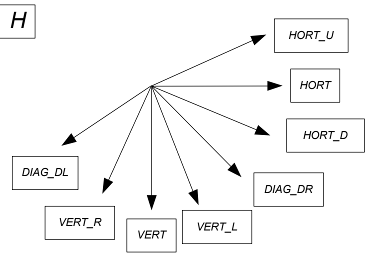


Fig. 1 (left) 4x4 block with elements (*a* to *p*) and neighbouring pixels (*A* to *Q*);
Fig. 2 (right) Eight direction-biased intra-modes in a 4x4 block.

The mode selected is that which produces the least Lagrangian cost. Since the H.264 standard employs a brute force algorithm to examine all nine modes, the computational burden is unavoidable. This increases the processing time of the encoder.

A number of suggestions have been made [3-6] to reduce the computational complexity of H.264 intra-frame coding. Meng et al [3] selected an optimal mode by computing a partial cost for down-sampled pixels instead of for a 4x4 block. However, this algorithm lacked the integration with Lagrangian optimisation. Pan et al [4] proposed a fast algorithm based on edge information. A recently-published improvement in [5] claims an average computational reduction of 60%. However, the picture distortion and bit rate overhead arising from incorrect predictions are relatively significant. Kim et al [6] advocated an algorithm featuring the joint spatial and Hadamard transform domain. The results demonstrated improvements in terms of picture quality and compression ratio.

In this paper, a fast intra-mode selection algorithm, Fintra, is developed. The success of the proposed algorithm is achieved by selecting fewer modes to undergo full Lagrangian cost evaluation. The entire selection process operates in the discrete cosine transform (DCT) domain. Furthermore, the complete computation of all DCT spectrums is not required, and the extrapolation processes are only performed for the modes that require further Lagrangian cost evaluation.

The remainder of this paper is organised as follows: Section 2 provides a detailed formulation of the proposed algo-

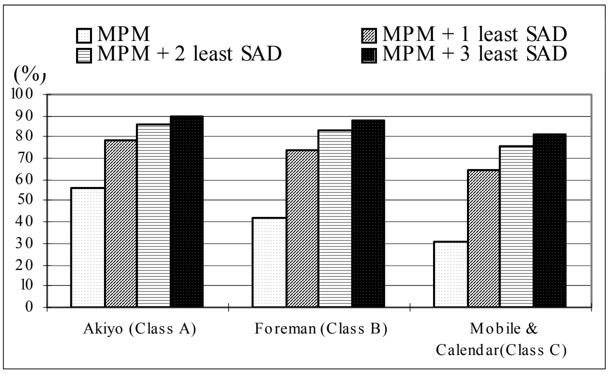


Fig. 3 Match percentage between two measurements: the least SAD cost and the least Lagrangian cost

rithm, the results of extensive simulations are summarised in Section 3, and finally, Section 4 draws some conclusions.

2. THE PROPOSED FINTRA ALGORITHM

Generally, the residues of intra blocks have a relatively large block energy due to the spatial prediction from neighbouring pixels. Consequently, it is more efficient to operate in the frequency domain rather than in the spatial domain. Furthermore, it is observed that the modes that provide the least residue energy will result in minimum rate R and hence minimise the Lagrangian cost. The chart in Fig. 3 illustrates this observation. The chart shows the relationship between the candidates selected by distortion cost and those by Lagrangian evaluation. Results were obtained by coding thirty frames of three test sequences in CIF format (352×288 pixels), Akiyo, Foreman, and Mobile & Calendar. The sequences represent three different degrees of spatial correlation and movement. 4x4 block intra-coding was used. The most probable modes (MPM), the candidates predicted from prior knowledge of neighbouring blocks, account for 56%, 42% and 30% of the mode decisions made by Lagrangian evaluation, for the three test sequences respectively. These percentages increase to 89%, 88% and 81% when three more candidates, each selected as having the lowest SAD values, are chosen.

To reduce the computational cost of expensive Lagrangian cost evaluation, we can limit the number of modes (to say M) that need to undergo the full evaluation process. The M modes are those with the least residue energy from amongst all the possible members. The following subsections describe the fast algorithm in detail.

2.1. Algorithm formulation

The selection criterion is the least residue energy which can be measured from the sum of absolute difference (SAD) of the DCT residue block. First, let us denote a 4x4 original block to be $\mathbf{B}$ and any predicted block to be $\mathbf{P}_{\text{mode}}$. The SAD of the DCT residue block is given by

$$\begin{aligned}\text{SAD}_{\text{DCT(residue)}} &= \sum_{m=0}^{3}\sum_{n=0}^{3}\left|\text{DCT}\{\mathbf{B}(m,n)-\mathbf{P}_{\text{mode}}(m,n)\}\right| \\ &= \sum_{m=0}^{3}\sum_{n=0}^{3}\left|\text{DCT}\{\mathbf{B}(m,n)\}-\text{DCT}\{\mathbf{P}_{\text{mode}}(m,n)\}\right| \\ &= \sum_{x=0}^{3}\sum_{y=0}^{3}\left|\text{DCT}_{\mathbf{B}}(x,y)-\text{DCT}_{\mathbf{P}|\text{mode}}(x,y)\right|\end{aligned}$$

Then, according to the definition of the DCT,

$$\begin{aligned}\text{SAD}_{\text{DCT(residue)}} \approx &\left|DC_{\mathbf{B}}-DC_{\mathbf{P}|\text{mode}}\right|+\left|AC_{\mathbf{B}}(1,0)-AC_{\mathbf{P}|\text{mode}}(1,0)\right| \\ &+....+\left|AC_{\mathbf{B}}(x,y)-AC_{\mathbf{P}|\text{mode}}(x,y)\right| \qquad (3)\end{aligned}$$

where DC represents the low frequency coefficient of a DCT block and $AC(x,y)$ accounts for the high frequency coefficient located at (x,y). Equation (3) shows that an approximation of $\text{SAD}_{\text{DCT(residue)}}$ can be made provided that enough AC coefficients are selected. It is observed that the low-frequency AC coefficients normally contain more energy than the high-frequency coefficients. Consequently, we select only a few low-frequency AC coefficients from the upper top triangle region of a DCT block, let's say, $AC(1,0)$, $AC(0,1)$ and $AC(1,1)$. Note that the accuracy of the approximation increases if more coefficients are involved.

By simple calculations from the 2D-DCT definition, we can easily obtain the formulae of the $DC_{\mathbf{B}}$ and $AC_{\mathbf{B}}(x,y)$ of a 4×4 block:

$$DC_{\mathbf{B}} = f_0 \times [r_{\mathbf{B}}(0)+r_{\mathbf{B}}(1)+r_{\mathbf{B}}(2)+r_{\mathbf{B}}(3)] \qquad (4)$$

$$AC_{\mathbf{B}}(1,0) = f_1 \times [r_{\mathbf{B}}(0)-r_{\mathbf{B}}(3)] + f_2 \times [r_{\mathbf{B}}(1)-r_{\mathbf{B}}(2)] \qquad (5)$$

$$AC_{\mathbf{B}}(0,1) = f_1 \times [c_{\mathbf{B}}(0)-c_{\mathbf{B}}(3)] + f_2 \times [c_{\mathbf{B}}(1)-c_{\mathbf{B}}(2)] \qquad (6)$$

$$\vdots \qquad\qquad \vdots$$

where f_1, f_2, f_3, are the scalar values 0.2500, 0.3267 and 0.1353, respectively. $r_{\mathbf{B}}(m)$ and $c_{\mathbf{B}}(n)$ represent the sum of the image intensities in the m^{th} row and n^{th} column of $\mathbf{B}$ respectively. For example, $r_{\mathbf{B}}(0) = a+b+c+d$ (refer to Fig. 1 for the definitions of a to d)

Next, we consider how to efficiently access the $\text{DC}_{\mathbf{P}|\text{mode}}$ value of the predicted block, $\mathbf{P}_{\text{mode}}$. Unlike the original block, the predicted blocks are the direction-biased extrapolations from neighbouring pixels. In order to formulate an efficient calculation, we represent (4) in matrix form with a' to p', the elements of any predicted block, as

$$DC_{\mathbf{P}|\text{mode}} = [\underbrace{f_0, \ldots\ldots\ldots, f_0}_{16}]_{1\text{x}16} \bullet \begin{bmatrix} a' \\ \vdots \\ p' \end{bmatrix}_{\text{mode}} \qquad (7)$$

In a similar manner, a matrix formula can be provided to relate the predicted elements and the neighbouring samples (A to Q):

$$\begin{bmatrix} a' \\ \vdots \\ p' \end{bmatrix}_{\text{mode}} = \mathbf{C}_{\text{mode}} \bullet \begin{bmatrix} A \\ \vdots \\ \vdots \\ Q \end{bmatrix}_{17\text{x}1} \qquad (8)$$

where $\mathbf{C}_{\text{mode}}$ is a 16-by-17 conversion matrix. For example, $\mathbf{C}_{\text{HORT}}$, the conversion matrix for the horizontal mode, extrapolates the horizontal pixel I to the elements in first row, i.e., *a*' to *d*'. All the coefficients in the first 4 rows of $\mathbf{C}_{\text{HORT}}$ are zero except for the ninth coefficients ($\mathbf{C}_{1,9}$, $\mathbf{C}_{2,9}$, $\mathbf{C}_{3,9}$, $\mathbf{C}_{4,9}$, i.e., position of *I*), which are one.

We then obtain the relationship between $DC_{\mathbf{P}|\text{mode}}$ and the neighbouring pixels (*A* to *Q*) by combining (7) and (8).

$$DC_{\mathbf{P}|\text{mode}} = \boldsymbol{\omega}_{\text{mode}} \bullet \begin{bmatrix} A \\ \vdots \\ \vdots \\ Q \end{bmatrix}_{17\times 1} \qquad (9)$$

where $\boldsymbol{\omega}_{\text{mode}}$ is a 1-by-17 transpose vector. By arranging the elements of $\boldsymbol{\omega}_{\text{mode}}$ to form a matrix, we can obtain the values of $DC_{\mathbf{P}|\text{mode}}$ for all the nine modes.

$$\begin{bmatrix} DC_{\mathbf{P}|HORT} \\ DC_{\mathbf{P}|VERT} \\ \vdots \\ DC_{\mathbf{P}|DIAG\_DL} \end{bmatrix}_{9\times 1} = \mathbf{\Omega}_{DC} \bullet \begin{bmatrix} A \\ \vdots \\ \vdots \\ Q \end{bmatrix}_{17\times 1} \qquad (10)$$

where

$$\mathbf{\Omega}_{DC} = \begin{bmatrix} \boldsymbol{\omega}_{HORT} \\ \boldsymbol{\omega}_{VERT} \\ \vdots \\ \boldsymbol{\omega}_{DIAG\_DL} \end{bmatrix}_{9\times 17} \qquad (11)$$

$\mathbf{\Omega}_{DC}$ in (11) is a 9-by-17 sparse matrix. $\mathbf{\Omega}_{AC(x,y)}$ can be deduced in a similar manner from (8) to (11) Note that $\mathbf{\Omega}_{DC}$ and all $\mathbf{\Omega}_{AC(x,y)}$ can be calculated and stored in advance.

2.2. Algorithm description

The proposed algorithm utilizes (3) to shortlist M candidates from the 9 modes. However, since the experimental investigations detailed in Fig. 3 indicate that the most probable mode (MPM) has a higher chance of being selected as an optimal mode, it is included in the short-listed candidates although it may not produce the least residue energy. The proposed algorithm is summarized as follows:

A1. Evaluate (4)-(6) to obtain *DC* and several *AC* coefficients for the original processed block

A2. Calculate values of DC_{mode} and the same number of $AC_{\text{mode}}(x,y)$ coefficients for the 9 predicted blocks, by utilizing $\mathbf{\Omega}_{DC}$ and $\mathbf{\Omega}_{AC(x,y)}$.

A3. Apply SAD evaluation (3) to shortlist M candidates with the smallest residue energies (including MPM).

A4. Select an optimal mode that minimizes (1) from the short-listed candidates.

The proposed fast algorithm employs the inherent frequency characteristic of an original block and its predicted block without any a priori knowledge, such as a predefined threshold. This feature is considered one of the main advantages of the proposed algorithm in that it is appropriate for all sequences, without modification. Furthermore, the utilisation of matrices, Ω_{DC} and $\Omega_{AC(x,y)}$ obtains the required frequency coefficients without an increase to the computational overhead.

3. SIMULATION RESULTS

This section compares the results of the proposed Fintra algorithm with the two best performing algorithms [5, 6] detailed in the literature. Each of the algorithms has been integrated with the JM6.1e software recommended by JVT [2] for performance evaluation. Results are presented as improvements over the rate-distortion optimisation version of the JM6.1e benchmark. Except were stated, the selected sequences are of QCIF resolution (176x144 pixels). Various values of M are chosen for the proposed Fintra algorithm to compare the performance with the two aforementioned algorithms.

3.1. Simulation results for M = 3

A performance comparison of the proposed Fintra algorithm with Kim et al's algorithm reported in [6] is presented. Both algorithms select 3 candidates (M=3) to undergo a full Lagrangian evaluation. In total 30 frames of each sequence were encoded with intra-frame coding only. All parameter settings for these two algorithms were identical. Comparisons are given for rate-distortion performance and computational speedup.

Fig. 4 and Fig. 5 exhibit two PSNR-rate relationship diagrams for the Coastguard and Foreman sequences. The solid, dashed, and dotted curves represent the JM6.1e benchmark,

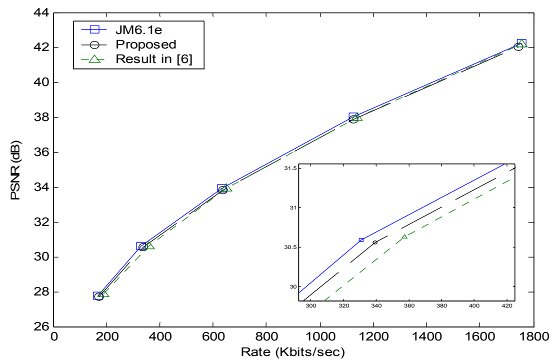


Fig. 4 Rate-distortion relationship diagram of three algorithms for the Coastguard sequence.

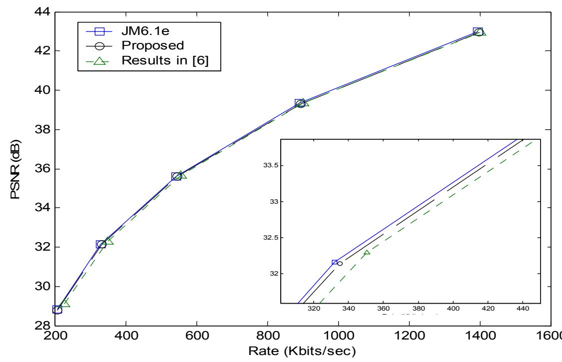


Fig. 5 Rate-distortion relationship diagram of three algorithms for the Foreman sequence.

the proposed Fintra algorithm, and the algorithm reported in [6], respectively. Generally, the PSNR-rate curves of both algorithms are virtually identical to that provided by JM6.1e software. The inlet diagrams in Fig. 4 and Fig. 5 show a magnified section of each of the curves. It is observed that the proposed algorithm reduces the PSNR by less than 0.10dB for any bit rate in the range. The PSNR degradation of Kim's algorithm [6] is approximately twice that our method. It is worth noting that human perception is especially sensitive at a PSNR of less than 36dB. Unnoticeable picture degradation is obtained by the proposed algorithm, as the average PSNR difference is less than the human perception threshold, widely recognised as 0.20dB. As to the speed up factor, both algorithms achieve similar computational savings in terms of the entire encoding time provided that the considerations of early termination thresholds and the simplified R-D model indicated in [6] are excluded. Computational savings of 61%-67% and 58%-63% are observed for the proposed algorithm and Kim et al's algorithm, respectively. The proposed Fintra algorithm appears to achieve a superior coding performance.

3.2. Simulation results for M = 2

The proposed Fintra algorithm is able to achieve a greater speedup when M is reduced. Sacrifices of other coding performance qualities, such as picture quality and compression ratio, are expected. This subsection discusses the situation when M=2. A performance comparison with the improved version of Pan et al's algorithm reported in [5] is presented. Since the implementation of Pan et al's algorithm uses the same JM software version, the results are directly taken from [5]. All parameter settings for both algorithms are identical. Comparisons are given for PSNR difference, bit rate difference, and speedup at a fixed Qp parameter.

Table 1 shows the three measurements for the two algorithms in comparison with the JM6.1e implementation. In terms of picture degradation and bit rate increase, the proposed Fintra algorithm provides an average of 0.08dB and 1.50% respectively. Pan et al reported [5] figures of 0.22dB and 3.57%. Note that PSNR is measured for a constant Qp parameter rather a fixed bit rate. The PSNR differences at a fixed bit rate are expected to be larger than those reported in Table 1. As to speedup, the proposed algorithm achieves a computational reduction of between 71% and 75%, whereas Pan et al [5] report a maximum of 65%. Table 1 indicates that the proposed algorithm outperforms the improved version of the algorithm reported in [5] in all respects.

4. CONCLUSIONS

In this paper, a fast algorithm operating in DCT domain to select an optimal mode for intra-frame coding is presented. The proposed Fintra algorithm utilises matrix formulae with low computational overhead to acquire the required frequency coefficients for short-listing candidates. Extensive simulations compare the performance of the proposed algorithm with existing approaches. The results verify the success of the proposed method. The rate-distortion curves show that the proposed algorithm achieves the same coding performance in terms of picture quality and compression ratio as that of the H.264/AVC standard, yet reduces the computational requirement by up to 75%.

REFERENCES

[1] "Information technology – coding of audio visual objects – Part 10: advanced video coding," ISO/IEC 14496-10:2003, Dec. 2003.

[2] JVT reference software, JM 6.1e, downloaded from http://bs.hhi.de/~suehring/tml.

[3] B. Meng, O. Au, C. Wang, and H. Lam, "Efficient intra-prediction mode selection for 4x4 blocks in H.264," *proceeding of ICME 2003*, pp. III-521 – III524, Baltimore, USA.

[4] F. Pan, X. Lin, S. Rahardja et al., "Fast mode decision for intra prediction," JVT-G013, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Mar. 2003, Pattaya, Thailand.

[5] F. Pan, X. Lin, S. Rahardja et al., "Fast intra mode decision algorithm for H.264/AVC video coding," *proceeding of ICIP 2004*, pp. 781– 784, Oct. 2004, Singapore.

[6] C. Kim, H. Shih, and C. Kuo, "Feature-based intra-prediction mode decision for H.264," *proceeding of ICIP 2004*, pp. 769 – 772, Oct. 2004, Singapore.

Sequences	PSNR difference		Bit rate difference		Speed up c.f. JM6.1e	
	Pan et al [5]	Proposed	Pan et al [5]	Proposed	Pan et al [5]	Proposed
Bus (CIF)	-0.22 dB	-0.12 dB	3.85 %	1.46 %	58.12 %	75.22 %
Coastguard	-0.11 dB	-0.06 dB	2.36 %	1.68 %	55.03 %	72.48 %
Container	-0.23 dB	-0.10 dB	3.70 %	0.77 %	56.36 %	71.86 %
Foreman	-0.29 dB	-0.07 dB	4.44 %	1.97 %	65.38 %	72.47 %
News	-0.29 dB	-0.09 dB	3.90 %	1.18 %	55.39 %	73.13 %
Paris (CIF)	-0.23 dB	-0.09 dB	3.21 %	1.47 %	57.78 %	73.43 %
Silent	-0.18 dB	-0.05 dB	3.54 %	2.00 %	65.17 %	71.86 %

Table 1 Coding performance and speedup compared with the algorithm of Pan et al [5], for fixed Qp.