LOW COMPLEXITY INTRA VIDEO CODING USING TRANSFORM DOMAIN PREDICTION

Chanyul Kim
Centre for Digital Video Processing,
Dublin City University, Ireland
email: chanyul.kim@eeng.dcu.ie

Noel E.O'Connor
Centre for Digital Video Processing,
Dublin City University, Ireland
email: oconnorn@eeng.dcu.ie

ABSTRACT
In this paper, a new low complexity intra coding framework is presented. The proposed method is extremely computationally efficient as it uses intra prediction in the DCT domain. To facilitate finding a good predictor, we propose to extend the number of neighouring blocks to be searched, based on a consideration of the type of edges we can expect to observe in the pixel data. The best predictor can be selected from the candidate blocks without recourse to rate-distortion optimisation or pixel interpolation. To obtain better performance we also propose to automatically adapt the entropy encoding block to the prediction mode used. Experimental results show that the encoding scheme compares favorably to H.264/AVC in terms of compression efficiency but with a significant reduction in overall computational complexity.

KEY WORDS
Intra video coding, low complexity, Intra prediction.

1 Introduction

Emerging consumer applications such as wireless video communications, wireless video cameras, disposable video cameras, and networked camcorders require low complexity encoders due to memory, computation, and power consumption limitations [1]. Many compression approaches are currently being investigated targeting such applications, including distributed video coding [2, 3] and low complexity hybrid video coding [4]. In distributed video coding, inter-frame coding based on motion estimation and compensation is performed at the decoder side in order to reduce encoder complexity. However, typically such approaches still use traditional intra coding schemes such as H.264/AVC intra coding and JPEG2000. Thus, intra coding is a potential bottleneck to achieving very low complexity video coding. Of course, beyond such advanced approaches, intra video coding is also widely used as coding method in its own right, particularly in surveillance video applications.

It is well known that the H.264/AVC standard achieves much higher coding efficiency than previous video coding standards such as MPEG-1/2/4 [5]. Moreover, H.264/AVC intra video coding has shown better quality compared to JPEG2000 [6]. The key feature of H.264/AVC intra coding is the prediction which is used to find spatial correlation. Although MPEG-4 uses a prediction method in the transform domain using the DC and several AC coefficients, its performance is not particularly good compared to H.264/AVC since new coding tools are deployed in H.264/AVC such as prediction based on pixel interpolation and rate distortion (RD) optimization. These kinds of spatial prediction coding methods are designed for reducing spatial redundancy by using the neighboring pixels of the current block. However, the resulting compression gain comes at the cost of a significant increase in processing time and memory accesses.

The most important and computationally complex functional block of intra coding is the prediction scheme employed. Therefore, much research has targeted reducing the prediction complexity. Feng *et al.* proposed a fast prediction mode in H.264/AVC based on edge pixels, where a Sobel edge operator was used to determine the edge direction [7]. Many trials to reduce prediction complexity can be found in [8, 9]. Unlike these approaches, the method reported in this paper does not use pixels but transform domain data. Junho *et al.* proposed a DCT-based prediction scheme where they defined an edge angle direction via the ratio of horizontal and vertical axis of DCT coefficients to obtain the prediction mode [10]. Their approach selected the prediction mode according to the angle direction and a DC threshold value. After selecting the prediction mode, subsequent processing is performed in the interpolated pixel domain as in H.264/AVC intra coding.

The objective of our proposed method is to obtain low complexity and reasonable quality compared to intra coding in the H.264/AVC standard. Thus, in this paper we are overly not concerned with compatibility with previous standards, *rather we are more interested in the investigation of a complete low complexity encoder framework*. In Section 2, H.264/AVC intra coding is discussed for context as it is the baseline against which we compare our technique. In Section 3, the proposed intra coding method is discussed in detail. This consists of two core contributions: the prediction method and a modified entropy encoding module. Experimental results are presented in Section 4 and conclusions in Section 5.

2 Intra Video Coding in H.264/AVC

Figure 1 shows the functional block of H.264/AVC intra coding. For our purposes, there are two issues to be considered in this scheme. One is the rate distortion (RD) optimized mode decision, corresponding to selecting 4×4 or 16×16 intra mode. The other is choosing the prediction method in the selected mode. For example there are nine modes for 4×4 prediction and four modes for 16×16 prediction, as shown in Figure 2.

The rate distortion calculation is defined as:

$$J_m^{\min} = \min\left(D_m + \lambda_m \times R_m\right) \tag{1}$$

Where J_m is the minimum Lagrange cost, D_m is the sum of squared distances between the original block and the reconstructed block for prediction mode m and the Lagrange multiplier $\lambda_m = 0.85 \times 2^{(qp)/3}$, where qp is the quantization factor (from 0 to 51). The prediction mode is determined as the value that minimizes J_m. After choosing the prediction mode, residual DCT coefficients of blocks are sent to a CAVLC[1] or CABAC[2] entropy encoder. In this process, the RD method based on Lagrange minimum cost has the highest computational cost as discussed in [11].

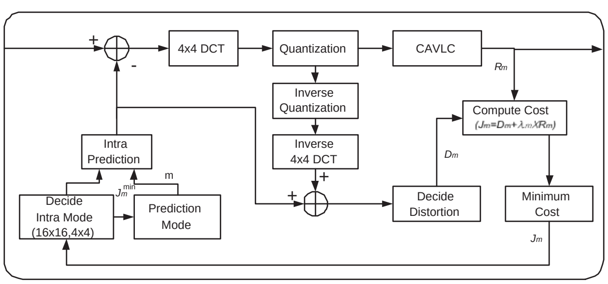
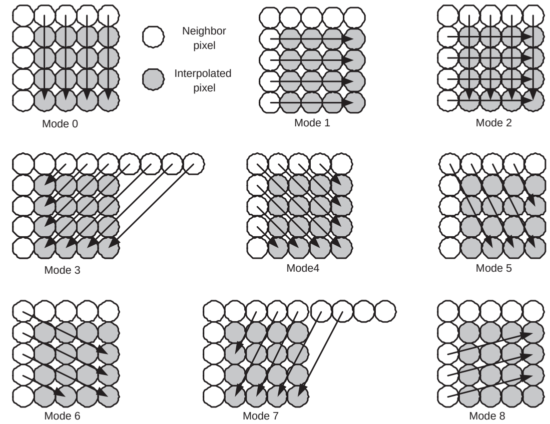


Figure 1. H.264/AVC Intra Coding function block

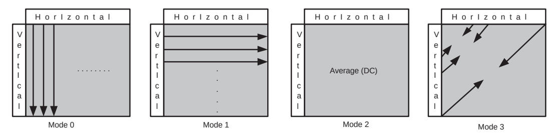
3 The Proposed Low Complexity Intra Video Encoding Architecture

The proposed low complexity intra encoding block, as illustrated in Figure 3, consists of Integer DCT, quantization, intra prediction and a modified CAVLC block. The 4×4 integer DCT, a standard tool in H.264/AVC and VC-1 [12] is used. The prediction block reduces spatial redundancy and has eight modes for both AC and DC coefficient prediction. A modified CAVLC module is used for entropy coding of the residual coefficients obtained via the prediction module. Depending on the prediction mode, an adaptive scan order is selected for reducing coefficient run lengths. The proposed method is motivated by reducing computational complexity within a reasonable quality degradation compared to H.264/AVC intra coding.

[1]Content Adaptive Variable Length Code
[2]Content Adaptive Binary Arithmetic Code



(a) 4×4 intra prediction mode



(b) 16x16 intra prediction mode

Figure 2. Intra prediction mode (a) 4×4 intra, (b) 16×16 intra

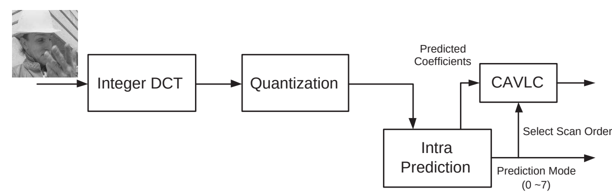


Figure 3. Low complexity encoder function block

3.1 The Proposed Prediction Scheme

In this section, the various steps in the prediction method, that is the core contribution to this paper, are described in detail. First, we motivate the definition of candidate predictors, then define how the prediction is calculated and finally describe how entropy encoding can be modified to reflect the chosen prediction.

3.1.1 Selecting Prediction Blocks

Our approach is based upon our observation that if the edge strength and direction remains the same between two blocks, then the difference (prediction residual) between the DCT coefficients of these blocks will be low. On the other hand, if there is a change in the edge characteristic, then the difference (prediction residual) will be substantially increased. This is illustrated in Figure 4 which shows the prediction residual of blocks predicted in the same mode in H.264/AVC. Thus, it seems intuitive to structure the search for candidate predictors among neighbour-

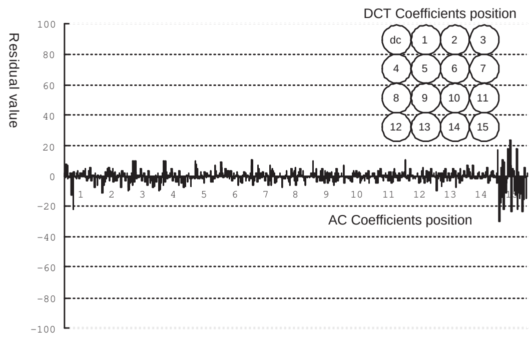


Figure 4. The histogram of predicted 4×4 blocks in H.264/AVC within the first frame of foreman sequence for residual DCT coefficients between blocks with the same selected prediction mode

ing blocks based on the types of edges we can expect to observe in a given block. Further, it indicates that we can simply difference the DCT coefficients of the current and candidate blocks in order to estimate how well a candidate predicts the block under consideration.

Considering H.264/AVC, it can be noted that there are nine modes for 4×4 block prediction (including a DC mode) and four modes for 16×16 block prediction, obtained by interpolation with neighboring pixels, see Figure 2. These predictions can be considered to be selected according to looking for continuity of edge directions corresponding to: $0°, \pm45°, 90°, \pm26.5°, \pm63.4°$. This indicates that in our scheme we should look in these directions for candidate predictors of whole blocks of DCT coefficients. For edge directions $0°, \pm45°, 90°$, only four neighbouring blocks need to be considered as potentially containing the continuation of the edge inteh current block, and thus likely to be good predictors. However, for $\pm26.5°, \pm63.4°$ directions, four more blocks are required as potential candidates for potentially continuing the edge.

This is explained as follows. If the edge direction of the current block is $0°, \pm45°, 90°$, the upper, left, upper left and upper right blocks can be considered as predictor blocks since the edge patten potentially does not change with respect to the current block for any of these positions. However, if the edge direction of the current block is $\pm26.5°$ or $\pm63.4°$, the required edge may appear in the other positions as shown in Figure 5. For example, if the edge direction is $23.5°$, then the same pattern will appear in $MB_{x+2,y-1}$. Therefore, eight 4×4 transformed blocks are used for determining the prediction mode. That is, $MB_{x-1,y}$,$MB_{x,y-1}$,$MB_{x-1,y-1}$,$MB_{x+1,y-1}$ are needed for horizontal (0°), vertical (90°), right diagonal (45°) and left diagonal ($-45°$) and $MB_{x+2,y-1}$, $MB_{x+1,y-2}$, $MB_{x-1,y-2}$, $MB_{x+2,y-1}$ are used for $26.5°$, $63.4°$, $-63.4°$ and $-26.5°$ edge directions respectively. It should be noted that since we use whole blocks of DCT coefficients all eight positions can be searched given the low

Mode	Block position
$P_0(AC, DC)$	$MB_{x,y-1}$
$P_1(AC, DC)$	$MB_{x-1,y}$
$P_2(AC, DC)$	$MB_{x-1,y-1}$
$P_3(AC, DC)$	$MB_{x+1,y-1}$
$P_4(AC, DC)$	$MB_{x-2,y-1}$
$P_5(AC, DC)$	$MB_{x-1,y-2}$
$P_6(AC, DC)$	$MB_{x+2,y-1}$
$P_7(AC, DC)$	$MB_{x+1,y-2}$
$P_8(AC)$	$index : P_0(AC) = P_1(AC)$

Table 1. Prediction Modes

complexity of this operation, so that we do not need to explicitly estimate the edge characteristic of the current block (although this is possible – see discussion in the Conclusions section).

3.1.2 Prediction Methods

The prediction modes available are outlined in Table 1. DC prediction consists of eight modes whereas AC prediction has nine modes. $P_m(AC)$ is calculated to find the index of the block with minimum absolute sum of difference value between the current block and neighboring blocks as specified in Equation 2. Since we have no knowledge about the edge of the current block, all DCT coefficients of neighboring blocks are compared to those of the current block. The prediction mode is selected as the minimum $P_m(AC)$. $P_m(DC)$ is calculated in the same way as $P_m(AC)$.

$$\begin{aligned}
&P_m(AC) = \arg\Big[\min\Big[\sum_{ac=1}^{ac=15} |MB_{x,y}^{DCT_{ac}} - MB_{x-i,y-j}^{DCTac}|\Big]\Big] \\
&P_m(DC) = \arg\Big[\min\Big[|MB_{x,y}^{DCT_{dc}} - MB_{x-i,y-j}^{DCT_{dc}}|\Big]\Big] \\
&\quad (i,j) \in \{(1,0),(0,1),(2,1),(1,1), \\
&\qquad (-1,1),(-2,1),(-1,-2),(1,2)\}
\end{aligned} \tag{2}$$

Where $P_m(AC), P_m(DC)$ are AC and DC prediction modes, $MB_{x,y}^{DCT_{ac}}$ is the DCT AC coefficients of the 4×4 block at (x, y) position and (i, j) is a position index.

After determining the prediction mode, residual DCT coefficients can be calculated as in Equation 3. $P_8(AC)$ indicates the index when $P_1(AC)$ and $P_0(AC)$ have the same minimum sum of absolute value which means DC prediction in H.264/AVC. In this mode, predicted DCT value can be obtained by interpolating upper block and left block to minimize prediction error as $R_8(AC)^{DCT_i}$ of Equation 3. The difference of DCT coefficients between the current block and predicted block is used for generating residual DCT coefficients in the rest of the modes. Residual DCT coefficients and prediction mode bits are sent to the decoder through the CAVLC function block.

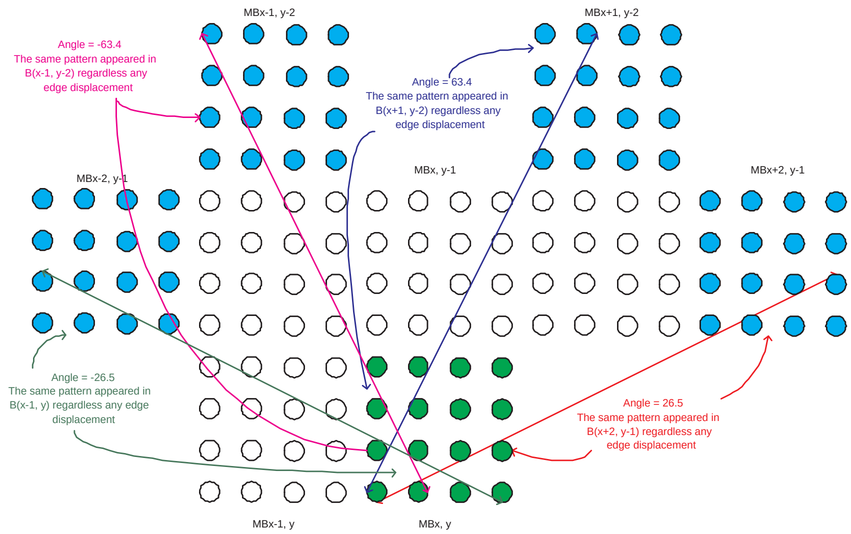


Figure 5. Candidate blocks used as predictors for current block. Additional blocks needed for considering edge angles=$\pm 26.5°$, $\pm 63.4°$ are shown in blue.

$$R_8(AC)^{DCT_i} = (MB_{x,y}^{DCT_i}) - (MB_{x-1,y}^{DCT_i} + MB_{x,y-1}^{DCT_i} + 1) >> 1$$

$$R_m(AC)^{DCT_i} = (MB_{x,y}^{DCT_i} - MB_{P_m(AC)}^{DCT_i}) \text{ if } m \neq 8 \quad (3)$$

Where $R_m(AC)^{DCT_i}$ is residual ith AC coefficients and $>>$ is shift operation.

3.2 Modified Content Adaptive Variable Length Code

After selecting the prediction mode, the residual coefficients between the current block and the predicted block are sent to the CAVLC function block. The CAVLC function is implemented using the same method as in H.264/AVC except for applying adaptive scan order according to the selected prediction mode. The concept of adaptive scan order was previously presented by Jie et.al [13]. The proposed method uses a similar method to their approach. The modified scan order is motivated based on the histogram of DCT coefficients, as illustrated in Figure 6, and gives advantages such as low memory access overhead. In our approach, the scan order is automatically selected according to the prediction mode $P_m(AC)$ in order to minimize run lengths as shown in Figure 7. If $P_0(AC)$ is selected as the prediction mode, the current block may have a vertical dominant edge. Therefore, the difference between two blocks also has vertical dominant edge components even though the value of components is reduced by prediction and thus we choose a vertical scan for $P_0(AC)$. In case $P_1(AC)$ is selected, then the horizontal scan order can be selected. The traditional zigzag scan order is used for all other prediction modes.

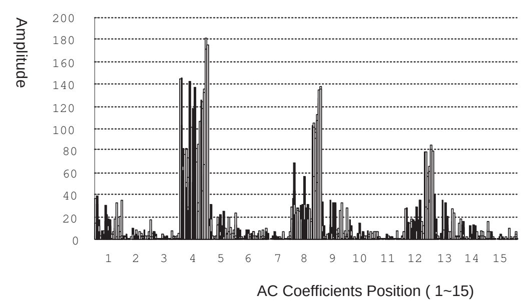


Figure 6. The histogram of DCT coefficients for mode 1 in H.264/AVC, indicating that the zig-zag scan may not be optimal for this mode.

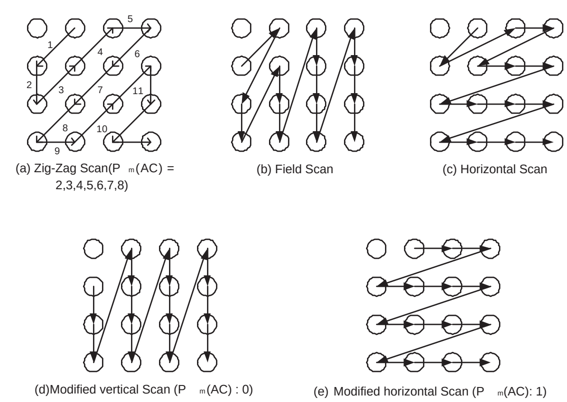


Figure 7. Scan order in CAVLC : (a) H.264/AVC standard method (b) & (c) Field scan and horizontal scan order [13] (d) & (e) Modified scan order

4 Implementation and Experimental Result

This section investigates the performance of the proposed low complex intra video coding method. The low complexity intra video coding framework is written in ANSI C++ and Intel Integrated Performance Primitive 5.3 library [14]. All tests were performed on an Intel Core(TM)2 Duo 3.6GHz with 2GB RAM using Window XP version 2002 with service pack 2. Our approach is compared with H.264/AVC reference software KTA1.6 based on JM 11.0 which was released in Jan. 2008 [15]. KTA has a more advanced feature set than H.264/AVC including adaptive quantization matrix selection, $\frac{1}{8}$ pel motion compensated prediction, adaptive prediction error etc. The proposed method focuses on intra coding, so only adaptive prediction error and adaptive matrix selection which affect intra video coding were considered. In order to ensure a fair test, only luminance coefficients were used for performance comparison, so U and V coefficients and intra prediction mode bits are disabled in the KTA software.

4.1 Comparison Rate Distortion Performance

Figure 8 shows the rate distortion performance obtained by our approach compared to KTA 1.6. In order to consider the effect of the rate distortion optimization routine in KTA reference software, we tested sequences with the rate-distortion function turned on and off. The Foreman, Hall Monitor, Mobile, Mother and Daughter sequences at CIF resolution are chosen as test sequences. From the results in Figure 8, it can be seen that the performance degradation with our approach is less than 0.8dB at the same bit rate in the Foreman and Hall sequences. However, the proposed method shows close to the same performance in Mobile and Mother and Daughter sequence. The proposed method appears to perform well for scenes with many edges (like Mobile) or scenes with sparse edges (like Mother and Daughter), but suffers a small performance drop for scenes with an edge distributions somewhere between these extremes (like Foreman and Hall Monitor). In addition, as the image quality gets worse due to quantization, the edges in the scene become simpler so the possibility of edges which are not members of the set $0°, 90°, \pm 26.5°, \pm 45° \pm 63.4°$ is reduced.

4.2 Profiling Execution Time

Table 2 shows the result of profiling, where execution time is measured in frames per second (fps). In KTA, rate distortion optimization (RDO) is the most computational consuming block. The proposed method increases the fps possible more than five times compared to H.264/AVC without the rate distortion optimization module turned on as shown in Table 2. This indicates very efficient operation.

Sequences	*RD on(off)*	*Proposed*	*Improvement*
foreman	*1.44 (12.47)*	*74.5*	*497%*
Hall monitor	*1.43 (12.15)*	*73.2*	*502%*
Mobile	*1.14 (11.49)*	*64.3*	*460%*
Mother	*1.52 (11.93)*	*75.8*	*535%*

Table 2. Average execution time for 30 intra frames compared to KTA 1.6. Unit is frame per second. Improvement figures quoted are with respect to RD turned off.

5 Conclusion and Consideration

In this paper, a new intra video coding framework is suggested. The quality performance is compared to the KTA1.6 reference software. The rate distortion performance is 0.8dB poorer than KTA for the Foreman and Hall Monitor sequences, however the performance is the same or better than that of the reference software for the Mobile and Mother and Daughter sequences. This implies that our approach is better for sequences exhibiting the extremes of spatial correlation. Since the prediction is performed in the transform domain, we don't need any feedback loop, rate distortion optimization or pixel interpolation. This reduces computational complexity in the encoder side. The complexity improvement is a factor of five compared to the reference software with the rate distortion optimization routine turned off.

In the future we plan to investigate whether or not explicitly characterizing the edge characteristics of blocks leads to even better performance in terms of both quality and reduced complexity. By estimating edge direction from DCT coefficients, it may be possible to cut down on the full search of candidate blocks as currently performed. This could have the twin benefits of reducing the search space for predictors whilst also obtaining better predictors.

Acknowledgement

The first author gratefully acknowledges the support for this research from Samsung Electronics Co., Ltd. and the second author is supported by Science Foundation Ireland under grant 03/IN.3/I361

References

[1] DISCOVER, "Distributed coding for video services," in *D19 Deliverable in DISCOVER PROJECT*, 2007.

[2] A. Aaron, Rui Zhang, and B. Girod, "Wyner-ziv coding of motion video," in *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, Nov. 2002, vol. 1, pp. 240–244.

[3] R. Puri and K. Ramchandran, "Prism: A new robust video coding architecture based on distributed

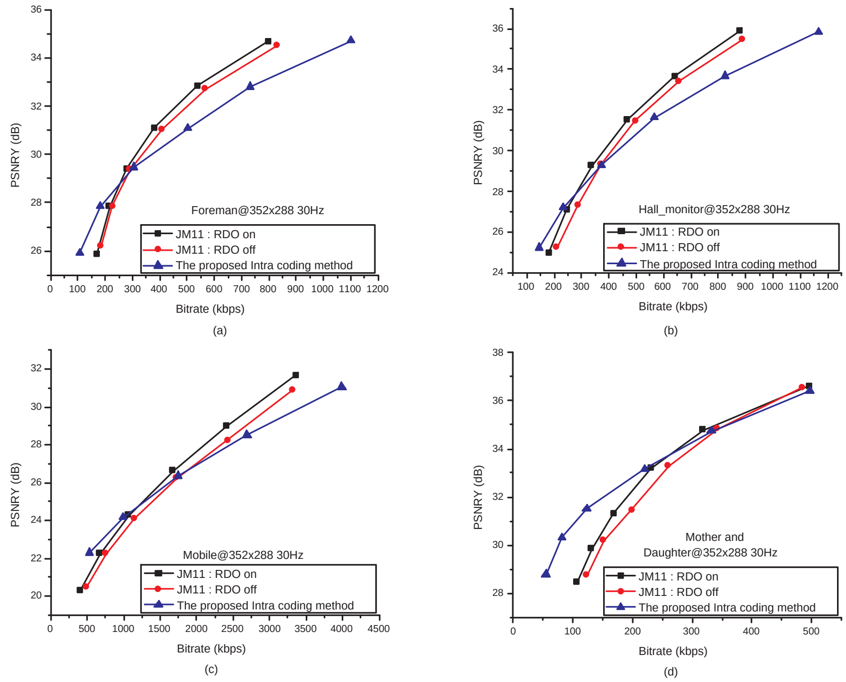


Figure 8. Rate Distortion result :(a) Foreman (b) Hall Monitor (c) Mobile (d) Mother and Daughter. All sequences are tested at 30Hz with CIF resolution.

compression principles," in *Proc. Allerton Conf*, Oct 2002.

[4] Yangli Wang and Chengke Wu, "Low complexity multiple description coding method for wireless video," in *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*, Mar. 2005, vol. 2, pp. 95–98.

[5] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the h.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[6] Detlev Marpea, Valeri Georgeb, Hans L. Cyconb, and Kai U. Barthelb, "Performance evaluation of motion-jpeg2000 in comparison with h.264/avc operated in pure intra coding mode," in *Proc. SPIE*, 2004, vol. 5266, pp. 129–137.

[7] Susanto Rahardja Keng Pang Lim Z. G. Li Dajun Wu Feng Pan, Xiao Lin and Si Wu, "Fast mode decision algorithm for intraprediction in h.264/avc video coding," in *IEEE Transactions on circuits and systems for video technology*, Jul 2005, vol. 15.

[8] Jong-Ho Kim, Byung-Gyu Kim, and Chang-Sik Cho, "Fast intra-mode prediction algorithm in h.264/avc video coding," in *IEICE TRANS. INF. & SYST*, Aug 2007, vol. E90-D.

[9] T.S. Cheng, C.C.and Chang, "Fast three step intra prediction algorithm for 4/spl times/4 blocks in h.264," in *Circuits and Systems*, 2005, vol. 2, pp. 1509–1512.

[10] Junho Jeong and David Nyeongkyu Kwon, "DCT based fast 4x4 intra-prediction mode selection," in *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*, Jan. 2007, pp. 332–335.

[11] Gerardo Fernández-Escribano, Pedro Cuenca, Luis Orozco-Barbosa, Antonio Garrido, and Hari Kalva, "Simple intra prediction algorithms for heterogeneous mpeg-2/h.264 video transcoders," *Multimedia Tools Appl.*, vol. 38, no. 1, pp. 1–25, 2008.

[12] SMPTE, *VC-1 Compressed Video Bitstream Format and Decoding Process*, SMPTE421M.

[13] Jie Jia, Eun-Ku Jung, and Hae-Kwang Kim, "Adaptive transform coefficient scan for h.264 intra coding," in *IEICE TRANS. INF. & SYST.*,, Oct 2007, vol. E90-D.

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

[15] *http://iphome.hhi.de/suehring/tml/download/KTA/*.