ORIGINAL PAPER

Linear-time prediction mode reduction method for intracoding in H.264/AVC

Kuo-Liang Chung[1] · Po-Chun Chang[1] · Wei-Ning Yang[1] · Wei-Jen Yang[2] · Chien-Hsiung Lin[1]

Received: 29 August 2016 / Accepted: 31 August 2016 / Published online: 16 September 2016

Abstract We present a linear-time elimination method to adaptively reduce the nine intraprediction modes in H.264/AVC to two, three, five, or nine modes while preserving a satisfactory reconstructed video quality. Resemblance between prediction modes and appearing frequency for each prediction mode are first determined in an off-line way from the training video sequences. Prediction modes are modeled as vertices and resemblance between two prediction modes as the weight of the connecting edge in a complete graph. Then, find the Hamilton cycle with the minimum sum of weights. When encoding each intrablock, prediction modes corresponding to the adjacent vertices along the Hamilton cycle are examined for similarity. The prediction mode which appears less frequently in a pair of similar prediction modes is declared as redundant and eliminated. When compared with Laroche et al.'s method which aims to reduce the bitrate, experimental results demonstrate that the proposed elimination method can substantially reduce the execution time while preserving the peak signal-to-noise ratio and bitrate performance.

Keywords Execution time, H.264/AVC · Rate-distortion optimization · Quality · Redundant intraprediction modes elimination

1 Introduction

H.264/advanced video coding (AVC) [1] established by the Joint Video Team (JVT) of ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) has become the state-of-the-art video coding standard to tackle a large number of video applications. H.264/AVC provides good features to enhance the coding efficiency. For example, for intercoding, it exploits seven variable block sizes and multiple reference frames in motion compensation; for intracoding, it applies four prediction modes to 16 × 16 block and nine prediction modes to 4 × 4 and 8 × 8 blocks. In this research, we focus on intraframe coding in H.264/AVC. For enhancing the bitrate performance by reducing the bitrate requirement in the intraframe coding, several methods, such as the prediction mode information inference approach [2], chroma subsampling [3], and transform-based coding [4], were developed.

Intraframe bitrate is mainly composed of the bit requirement for recording the optimal prediction mode, which depends on the number of prediction mode candidates, and the resulting residuals. Most methods reduce the bitrate by substantially reducing the resulting residuals via increasing the number of prediction mode candidates. One alternative is to reduce the number of prediction mode candidates at little cost of increasing the residuals. However, under the low-bitrate environment, it is difficult to substantially reduce the residuals by increasing the number of prediction mode candidates, implying that we should reduce the

Supported by the National Science Council under the contract MOST 104/105-2221-E-011-004/118-MY3.

✉ Kuo-Liang Chung
klchung01@gmail.ntust.edu.tw

[1] Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei, Taiwan 10672

[2] Department of Information Management, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei, Taiwan 10672

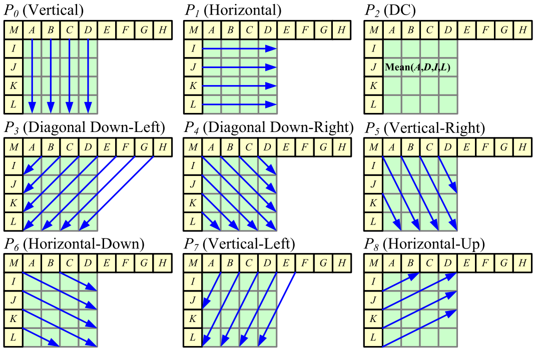

Fig. 1 Nine prediction modes of an intra block in the intra 4 × 4 partition mode

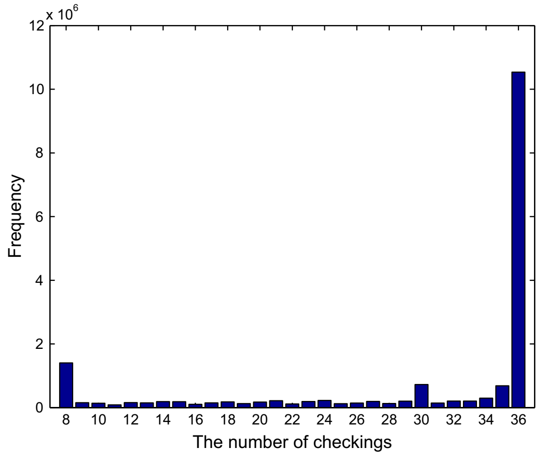

Fig. 2 The distribution of the number of similarity checkings in Laroche et al.'s elimination scheme

bitrate by decreasing the number of prediction mode candidates. Laroche et al. [2] proposed an $O(n^2)$-time method to eliminate redundant prediction modes and hence reduce the number of prediction mode candidates by exploiting the similarity between prediction modes, as shown in (Fig. 2), where n denotes the number of prediction modes.

In this paper, for each intrablock, we propose an $O(n)$-time method which adaptively reduces the nine intraprediction modes in H.264/AVC to two, three, five, or nine modes while preserving a satisfactory reconstructed video quality. The resemblance between any two prediction modes and the appearing frequency for each prediction mode are first determined based on the training video sequences. A complete graph is constructed where the prediction modes are modeled as vertices and the weight on each edge represents the resemblance between the two prediction modes connected by the edge. Then, find the Hamilton cycle with the minimum sum of weights. When encoding each intrablock of a new video sequence, prediction modes corresponding to the adjacent vertices along the Hamilton cycle are examined for similarity. The prediction mode which appears less frequently in a pair of similar prediction modes is eliminated. The proposed elimination scheme can reduce the bitrate by decreasing the number of prediction mode candidates and preserve the quality of the reconstructed video sequences. The computational complexity of the proposed elimination method is $O(n)$-time when compared to Laroche et al.'s $O(n^2)$-time method.

2 Intraprediction modes of H.264/AVC

In the intracoding of H.264/AVC, each 16 × 16 macroblock can be partitioned into a set of intrablocks of one of the three possible sizes, namely 16 × 16, 8 × 8, and 4 × 4; that is, the macroblock can be partitioned according to the intra 16 × 16 partition mode, the intra 8 × 8 partition mode, and the intra 4 × 4 partition mode.

Since the intra 16 × 16 partition mode only involves four intraprediction modes, it is not worth eliminating the redundant intraprediction modes in terms of the bitrate reduction. We primarily study the elimination for the intra 8 × 8 and 4 × 4 partition modes, each with the same nine intraprediction modes. The nine prediction modes for the intra 4 × 4 partition mode are shown in Fig. 1, where P_0, P_1, ..., and P_8 denote the eight directional prediction modes and one DC prediction mode. The prediction mode predicts, by extrapolation, each pixel value in an intrablock, and the resulting prediction errors are called residuals. The residuals are then quantized by a quantization matrix, which depends on the quantization parameter (QP), for encoding and the quantization usually leads to distortions when reconstructing the intrablock at the decoding side. For each intrablock, the encoder selects the optimal prediction mode P_{i^*} such that the rate-distortion (RD) cost is minimized; that is,

$$i^* = \arg\min_{i \in J}\{D_i + \lambda R_i\}, \quad (1)$$

where D_i and R_i represent, respectively, the distortion, and the bitrate corresponding to the prediction mode P_i; λ is the QP-dependent Lagrangian multiplier, which is a commensurate constant between distortion and bitrate, and $J = \{0, 1, 2, \ldots, 8\}$. The optimal prediction mode is then used for encoding and decoding the intrablock. In addition, the intracoding of H.264/AVC also supports a so-called *most probable prediction mode*, whose index is the smaller index of the optimal prediction modes corresponding to the upper and left intrablocks relative to the current intrablock. If the most probable prediction mode occurs to be the optimal prediction mode for the current intrablock, only one bit will be used for encoding the index of the optimal prediction mode; otherwise, four bits will be used, resulting in higher bitrate.

3 Survey of Laroche et al.'s method

In this section, we briefly introduce Laroche et al.'s $O(n^2)$-time method [2] for eliminating the redundant prediction modes. Their method is performed on all the partitioned 4×4 and 8×8 intrablocks. Since both 4×4 and 8×8 intrablocks employ the same elimination scheme, we only present the elimination scheme for the 4×4 intrablocks. Denote by $\mathbb{V} = \{\mathbf{B}^k | 1 \le k \le N\}$ the video sequence with N intrablocks, where $\mathbf{B}^k$ represents the k-th 4×4 intrablock. Let $\widehat{\mathbf{B}}_i^k$ denote the corresponding extrapolated intrablock for $\mathbf{B}^k$ using prediction mode P_i with $i \in J$. To eliminate the redundant prediction modes for $\mathbf{B}^k$, Laroche et al.'s method first determines the similarity between prediction modes P_i and P_j for $\mathbf{B}^k$ with $i \neq j$ and $i, j \in J$. P_i and P_j are similar for $\mathbf{B}^k$, denoted by $P_i^k \approx P_j^k$, if the following equality holds:

$$\mathcal{Q}^{-1}(\mathcal{Q}(\mathcal{DCT}(\widehat{\mathbf{B}}_i^k - \widehat{\mathbf{B}}_j^k))) = 0, \tag{2}$$

where $\mathcal{DCT}(\widehat{\mathbf{B}}_i^k - \widehat{\mathbf{B}}_j^k)$ denotes the discrete cosine transform (DCT) coefficient matrix of the difference between $\widehat{\mathbf{B}}_i^k$ and $\widehat{\mathbf{B}}_j^k$, $\mathcal{Q}$ the quantization matrix which depends on the QP used, and $\mathcal{Q}^{-1}$ the inverse quantization matrix. Note that the similarity condition test for Eq. (2) can be realized by adopting the method in [5]. Thus, once $P_i^k \approx P_j^k$, P_i or P_j can be eliminated from J. It is obvious that larger number of checkings for elimination is required for the cases of lower QPs. When there exist no similar prediction modes, it requires 36 checkings for elimination in Step 2, whereas only 8 checkings are required if all pairs of prediction modes are similar.

Let J^* denote the index set after eliminating the redundant prediction modes. Once J^* is determined by Laroche et al.'s method, an optimal prediction mode P_{i^*} is determined by Eq. (1), where J is replaced by J^*, for encoding. Since the size of J^* is usually smaller than the size of J, encoding the index of the optimal prediction mode selected from J^* requires fewer than four bits, which are used in H.264/AVC standard, and hence results in less bitrate. In summary, when compared with H.264/AVC, the advantage of less bitrate under the similar reconstructed quality is obtained at the cost of larger execution time and losing the H.264/AVC compliance in Laroche et al.'s elimination method.

4 The proposed linear-time redundant prediction mode elimination method

This section presents the proposed linear-time redundant prediction mode elimination method.

4.1 Determination of Hamilton cycle and appearing frequency for each prediction mode

Recall that $\widehat{\mathbf{B}}_i^k$ denotes the extrapolated intrablock k using prediction mode P_i. For each pair of P_i and P_j, calculate the average absolute difference between the corresponding extrapolated intrablocks by

$$\bar{D}_{i,j} = \frac{\sum_{k=1}^{N} \sum_{0 \le u,v \le 3} |\widehat{\mathbf{B}}_i^k(u,v) - \widehat{\mathbf{B}}_j^k(u,v)|}{N}, \quad 0 \le i, j \le 8,$$

where N represents the total number of intrablocks in the training video sequences and $\widehat{\mathbf{B}}_i^k(u,v)$ denotes the pixel value at position (u,v) in the extrapolated intrablock $\widehat{\mathbf{B}}_i^k$. The average absolute difference $\bar{D}_{i,j}$ reflects the resemblance between prediction modes P_i and P_j in the training video sequences. Smaller values of $\bar{D}_{i,j}$ indicate that prediction modes P_i and P_j are more resembling in the training video sequences. Computed from four training VCEG video sequences [6] with mixed spatial resolutions, Table 1 gives the resemblance matrix for each pair of prediction modes.

Table 1 Resemblance matrix for pairs of prediction modes based on the training VCEG video sequences

	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
P_0	0.0	357.7	209.4	200.9	228.3	154.1	290.0	136.7	376.5
P_1	357.7	0.0	210.3	388.7	237.3	297.6	162.0	369.8	109.2
P_2	209.4	210.3	0.0	249.5	158.8	166.8	166.4	217.7	216.4
P_3	200.9	388.7	249.5	0.0	289.7	247.3	333.3	100.5	402.7
P_4	228.3	237.3	158.8	29.65	0.0	119.8	121.5	258.7	267.2
P_5	154.1	297.6	166.8	247.3	119.8	0.0	210.0	206.4	320.6
P_6	290.0	162.0	166.4	333.3	121.5	210.0	0.0	309.1	207.3
P_7	136.7	369.8	217.7	100.5	258.7	206.4	309.1	0.0	385.8
P_8	376.5	109.2	216.4	402.7	267.2	320.6	207.3	385.8	0.0

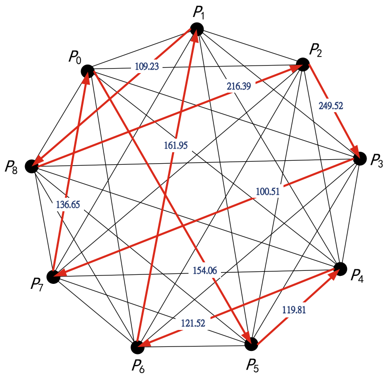

Fig. 3 The constructed complete undirected weighted graph and the Hamilton cycle with the minimum sum of weights

A complete undirected weighted graph, as depicted in Fig. 3, is constructed by modeling prediction modes as vertices and resemblance between two prediction modes as the weight on the edge which connects the prediction modes. Finding the sequence of prediction modes with the maximum overall resemblance is equivalent to finding the Hamilton cycle with the minimum sum of weights. That is, to find the sequence of prediction modes with the maximum overall resemblance, we first construct the complete undirected weighted graph $G = (V, E, w)$, where $V = \{P_i | 0 \leq i \leq 8\}$ and $E = \{(P_i, P_j) | 0 \leq i < j \leq 8\}$ are, respectively, the sets of vertices and edges, and $w = \{w(P_i, P_j) = \bar{D}_{i,j} | 0 \leq i < j \leq 8\}$ denotes the set of weights on the edges. Then, find the shortest tour of G, namely the Hamilton cycle $H = P_{\pi(0)} \rightarrow P_{\pi(1)} \rightarrow \ldots \rightarrow P_{\pi(8)} \rightarrow P_{\pi(0)}$, where π is a permutation of vertices such that $\sum_{i=0}^{7} w(P_{\pi(i)}, P_{\pi(i+1)}) + w(P_{\pi(8)}, P_{\pi(0)})$ is minimized.

In graph theory, finding the Hamilton cycle in a graph is an NP-complete problem [7]. Since there are only nine prediction modes for each intrablock, there exist 8! possible Hamilton cycles, implying that the Hamilton cycle with the minimum sum of weights can be determined in 8! exhaustive enumerations. Based on the resemblance matrix in Table 1, the Hamilton cycle with the minimum sum of weights found by exhaustive enumeration is $P_1 \rightarrow P_8 \rightarrow P_2 \rightarrow P_3 \rightarrow P_7 \rightarrow P_0 \rightarrow P_5 \rightarrow P_4 \rightarrow P_6 \rightarrow P_1$, as shown by the red arrows in Fig. 3.

The Hamilton cycle with the minimum sum of weights corresponds to the sequence of prediction modes with maximum overall resemblance, indicating that the neighboring prediction modes along the Hamilton cycle are more resembling and one of the neighboring prediction modes may be redundant. Thus, contrasted to Laroche et al.'s method, only the neighboring prediction modes are examined for similarity in the proposed method. If the neighboring prediction modes are similar, the prediction mode which appears less frequently is regarded as redundant and will be eliminated. Computed from the training VCEG video sequences, Table 2 gives the appearing frequency for each prediction mode.

Table 2 Appearing frequency for each prediction mode based on the training VCEG video sequences

Mode	P_1	P_0	P_8	P_2	P_4	P_6	P_7	P_3	P_5
Frequency (%)	21.0	17.0	13.6	10.2	9.3	8.5	7.4	6.7	6.4

4.2 Elimination of redundant prediction modes

Since two adjacent prediction modes along the Hamilton cycle are resembling, there may exist one redundant prediction mode for elimination. The prediction mode which appears less frequently in a pair of similar prediction modes is declared as redundant and eliminated.

Denote by $P_{\pi(0)} \rightarrow P_{\pi(1)} \rightarrow P_{\pi(2)} \rightarrow P_{\pi(3)} \rightarrow P_{\pi(4)} \rightarrow P_{\pi(5)} \rightarrow P_{\pi(6)} \rightarrow P_{\pi(7)} \rightarrow P_{\pi(8)} \rightarrow P_{\pi(0)}$ the Hamilton cycle with the minimum sum of weights, where $P_{\pi(0)}$ is the prediction mode with the highest appearing frequency. Let f_i represent the appearing frequency of prediction mode P_i for $i = 0, 1, \ldots, 8$. Recall that, for intrablock $\mathbf{B}^k$, prediction modes are said to be similar, denoted by $P_i^k \approx P_j^k$, if the similarity condition test in Eq. (2) is satisfied. Since adjacent intrablocks tend to have similar texture, the most probable prediction mode is likely to be the optimal prediction mode when considering all the prediction modes and so should not be eliminated. Starting with $P_{\pi(0)}$, the proposed scheme for eliminating the redundant prediction modes for intrablock $\mathbf{B}^k$ can be described as follows.

Step 1: Let $J = \{0, 1, \ldots, 8\}$, $b = 0$, and $i = 1$.
Step 2: If $P_{\pi(b)}^k \approx P_{\pi(i)}^k$, go to Step 3; otherwise, let $b = i$ and go to Step 4.
Step 3: If $f_{\pi(b)} > f_{\pi(i)}$, eliminate $\pi(i)$ from J; otherwise, eliminate $\pi(b)$ from J and let $b = i$.
Step 4: Let $i = (i + 1) \bmod 9$. If $i = 1$, then go to Step 5; otherwise, go to Step 2.
Step 5: Include the index of the most probable prediction mode in J if it is not in J and stop.

Denote by $\tilde{J}$ the output index set after eliminating the redundant prediction modes. Since each step in the above procedure needs $O(1)$ time, the elimination scheme has $O(n)$-time complexity with n denoting the number of prediction modes. Furthermore, the same Hamilton cycle, determined in an off-line way, is used for examining similar prediction modes in each intrablock. Thus, for eliminating the redundant prediction modes in each intrablock, the computational complexity of the proposed method is $O(n)$.

4.3 Encoding and decoding the index of the optimal prediction mode

Once $\tilde{J}$ is obtained, the optimal prediction mode P_{i^*} is determined such that the RD cost is minimized; that is, $i^* = \arg\min_{i\in\tilde{J}}\{D_i + \lambda R_i\}$.

If the most probable prediction mode turns out to be P_{i^*}, then only one bit with value 1 is used to record the index of P_{i^*}. Since $\tilde{J}$ is different from the standard index set, which includes all the prediction modes, in H.264/AVC, we need to design a different codebook for identifying the index of the optimal prediction mode. Denote by $|\tilde{J}|$ the cardinality of $\tilde{J}$. If $|\tilde{J}| = 2$ and the most probable prediction mode is not the optimal prediction mode, then only one bit with value 0 is used to record the index of P_{i^*}. If $|\tilde{J}| > 2$ and the most probable prediction mode does not turn out to be the optimal prediction mode, we need $1 + \lceil\log_2(|\tilde{J}| - 1)\rceil$ bits to record the index of P_{i^*}.

Note that different sizes of $\tilde{J}$ may require the same number of bits for recording the index of P_{i^*} and hence result in the same recording cost in terms of the number of bits. For example, given that the most probable prediction mode is not the optimal prediction mode, the cases of $|\tilde{J}| = 6$, $|\tilde{J}| = 7$, $|\tilde{J}| = 8$, and $|\tilde{J}| = 9$ all require a four-bit codeword to record the index of P_{i^*}. Provided not increasing the number of bits for recording the index of P_{i^*}, we may augment $\tilde{J}$ to J^*, for further reducing the prediction errors, by adding back some eliminated prediction modes with higher appearing frequencies. That is, for further reducing the prediction errors while keeping the bit length of the codeword unchanged, if $|\tilde{J}| = 6$, $|\tilde{J}| = 7$, $|\tilde{J}| = 8$, or $|\tilde{J}| = 9$, we augment $\tilde{J}$ to J^* with $|J^*| = 9$; if $|\tilde{J}| = 4$ or $|\tilde{J}| = 5$, we augment $\tilde{J}$ to J^* with $|J^*| = 5$; if $|\tilde{J}| = 3$, we have $J^* = \tilde{J}$ with $|J^*| = 3$; if $|\tilde{J}| = 1$ or $|\tilde{J}| = 2$, we augment $\tilde{J}$ to J^* with $|J^*| = 2$. Table 3 summarizes the codebook design for J^*. Excluding the index of the most probable prediction mode, the prediction mode indices in J^* are listed in an ascending order. Given J^*, the codeword can be used to identify the optimal prediction mode. For example, suppose $J^* = \{0, 2, 5, 6, 8\}$ with element 5 denoting the index of the most probable prediction mode of the current intrablock and a codeword of length 3 is transferred to the decoder. The decoder first identifies the most probable prediction mode which is P_5. If the first bit of the codeword received by the decoder is 1, then P_5 is the optimal prediction mode of the current intrablock. If the first bit of the codeword is 0, the decoder first reconstructs J^* and then removes the index 5 of the most probable prediction mode from J^*. Finally, the last two bits of the codeword are used to determine the position, in the remaining J^*, of the index of the optimal prediction mode. For example, codeword 001 leads to the optimal prediction mode P_2, codeword 010 to P_6, and codeword 011 to P_8.

Table 3 The codebook design for J^*

$\|J^*\|$	J^*	Codeword
2	Most probable prediction mode index	1
	The other prediction mode index	0
3	Most probable prediction mode index	1
	The other two prediction mode indices	00
		01
5	Most probable prediction mode index	1
	The other four prediction mode indices	000
		001
		010
		011
9	Most probable prediction mode index	1
	The other eight prediction mode indices	0000
		0001
		0010
		0011
		0100
		0101
		0110
		0111

Return to the encoding side. The proposed scheme first eliminates redundant prediction modes, includes the most probable prediction mode if necessary, and then increases the size of the index set, by adding back the indices of the eliminated prediction modes with higher appearing frequencies, to the nearest larger integer of values 2, 3, 5, and 9. Finally, the optimal prediction mode P_{i^*} is determined by $i^* = \arg\min_{i\in J^*}\{D_i + \lambda R_i\}$.

To retrieve the optimal prediction mode at the decoding side, we first check the first bit of the codeword to determine whether the most probable prediction mode is the optimal prediction mode. If the first bit of the codeword occurs to be “1,” then the most probable prediction is the optimal prediction mode and the intrablock can be reconstructed. If the first bit of the codeword occurs to be “0,” first obtain J^* by the same elimination and augmentation schemes based on the reconstructed adjacent reference blocks and then identify the optimal prediction mode according to the codeword with $1 + \lceil\log_2(|J^*| - 1)\rceil$ bits.

4.4 Detailed steps for encoding and decoding procedures

This subsection describes the detailed steps of encoding and decoding for the index of the optimal prediction mode for each intrablock.

The detailed steps for encoding the optimal prediction mode for each intrablock can be summarized as follows.

Step 1: Obtain the reduced prediction mode index set $\tilde{J}$ by applying the proposed elimination scheme, which eliminates the redundant prediction modes along the Hamilton cycle.

Step 2: Augment the prediction mode index set from $\tilde{J}$ to J^*.

Step 3: Select from J^* the index of the optimal prediction mode which has the minimum RD cost.

Step 4: Construct the codeword for the optimal prediction mode index. If the most probable prediction mode occurs to be the optimal prediction mode, then the codeword is "1;" otherwise, the codeword starts with "0" and the sequence of the following binary bits indicates the ordinal number of the optimal prediction mode in J^* excluding the most probable prediction mode.

The detailed steps for retrieving, at the decoding side, the index of the optimal prediction mode can be summarized as follows.

Step 1: If the first bit of the codeword is "1," then the most probable prediction mode is the optimal prediction mode for the current intrablock and stop; otherwise, go to Step 2.

Step 2: Obtain the reduced prediction mode index set $\tilde{J}$ by applying the proposed elimination scheme, which eliminates the redundant prediction modes along the Hamilton cycle.

Step 3: Augment the prediction mode index set from $\tilde{J}$ to J^*.

Step 4: According to the codeword, determine the ordinal number of the optimal prediction mode in J^* excluding the most probable prediction mode.

5 Experimental results

For comparison, two testing VCEG video sequences [8,9] in CIF format with resolution 1080p were adopted for H.264/AVC and four compared methods including the proposed method, Laroche et al.'s method, Kwon et al.'s method [10], and the simple elimination (SE) method which uses only the five most frequent prediction modes in Table 2. In addition, another four training VCEG video sequences [8,9] were adopted to construct the resemblance matrix, the appearing frequency table, and the Hamilton cycle, as shown in Tables 1-2 and Fig. 3, respectively, for the proposed method. In the experiments, the VCEG test common conditions which only contain the High profile were adopted and all the frames in the sequences were intracoded. Furthermore, the context-based adaptive binary arithmetic coding (CABAC) was used as the entropy coding option. For comparison, four different QPs, 22, 27, 32, and 37 were considered in the experiments. All compared methods were implemented on the IBM compatible computer with Intel Core i7-3770 CPU 3.4 GHz and 8GB RAM. The operating system used was Microsoft Windows 7, and the implementation software platform was JM 18.5 [11], which is realized by Visual C++ 2012.

The PSNR of a reconstructed video sequence with K frames of size $X \times Y$ is expressed as

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{KXY} \sum_{k=1}^{K} \sum_{x=1}^{X} \sum_{y=1}^{Y} \left[F^k(i,j) - F'^k(i,j) \right]^2}$$

where $F^k(i, j)$ denotes the color value of the pixel at position (i, j) for the kth image frame in the original video sequence and $F'^k(i, j)$ is the analog for the reconstructed video sequence. Denote by N_b the bit number and N_f the frame number of the video sequence. Let H represent the frame rate which is the number of frames played per second. The bitrate of a reconstructed video sequence is defined as

$$\text{bitrate} = \frac{N_b H}{N_f}.$$

In general, the rate-distortion (RD) curves, which represent the values of PSNR against different values of bitrate, measure the trade-off between the quality and the storage for each compared method. Figure 4 contains the five RD curves of the four compared methods and H.264/AVC with the High profile for two test video sequences. When fixing the values of PSNR, the proposed and Laroche et al.'s methods require less bitrates than the other two methods for higher QPs, as shown in the middle row of Fig. 4. This is mainly because the bitrate for the redundant prediction modes is effectively eliminated from the compressed video sequences by the proposed and Laroche et al.'s methods. However, for the cases of lower QPs, since the bitrate resulted from the intraprediction only takes low proportion in the total bitrate, as depicted in Fig. 1 of [2], the proposed and Laroche et al.'s methods yield less gain in bitrate reduction, as shown in the bottom row of Fig. 4. The SE and Kwon et al.'s methods usually deliver inferior performance of PSNR and bitrate since the two methods eliminate the prediction modes without considering the similarity among the prediction modes.

Furthermore, based on the RD curves shown in Fig. 4, we computed the relative gains of the four compared methods against H.264/AVC. The relative gains can be measured by Bjøntegaard delta PSNR (BD-PSNR) [12], which is the average gain in PSNR over a range of bitrate. Table 4 gives the values of BD-PSNR of the four compared methods against H.264/AVC for two test video sequences with different spatial resolution formats. Positive values of BD-PSNR indicate that the method with the corresponding elimination

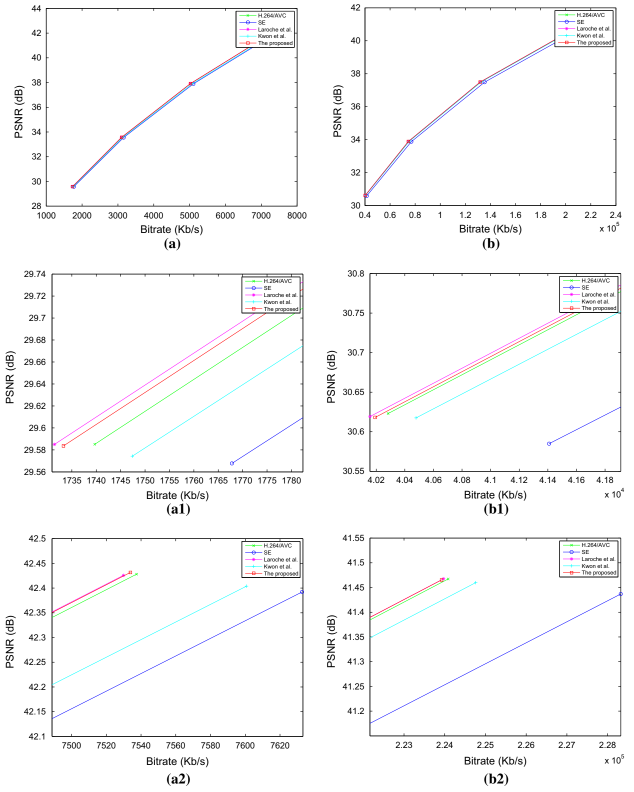

Fig. 4 The RD *curves* for the four compared methods and H.264/AVC corresponding to the two test video sequences with the High profile: (a-a2) Tempete (CIF) and (b-b2) CrowdRun (1080p). *Middle* and *bottom rows*, respectively, contain the corresponding magnified figures at lower and higher levels of bitrate

Table 4 BD-PSNR values (dB) over QP values {22, 27, 32, 37} for the proposed method, Laroche et al.'s method, the SE method, and Kwon et al.'s method against H.264/AVC for the two testing video sequences with different spatial resolution formats

High profile			
Format	Video sequence	Method	BD-PSNR
CIF	Tempete	Proposed	0.01
		Laroche et al.'s	0.02
		SE	−0.15
		Kwon et al.'s	−0.08
1080p	CrowdRun	Proposed	0.01
		Laroche et al.'s	0.01
		SE	−0.18
		Kwon et al.'s	−0.03

scheme delivers higher average PSNR, i.e., better quality of the reconstructed video sequences than that of H.264/AVC. From Table 4, it is clear that the proposed and Laroche et al.'s methods produce approximately zero values of BD-PSNR, indicating that the quality of the reconstructed video sequences of both the two methods is similar to that of H.264/AVC, and hence, no substantial quality degradation is incurred from the corresponding elimination schemes. In contrast, the SE method and Kwon et al.'s method produce negative values of BD-PSNR since the two methods may eliminate non-redundant prediction modes for not considering the similarity among prediction modes.

In summary, since the proposed method eliminates only the redundant prediction modes, the positive values of BD-PSNR confirmed that the proposed method can effectively reduce the bitrate while preserving the quality of the reconstructed video sequences. In addition, we also performed the experiments using the GOP structure to show the clear effect on bitrate reduction by the proposed method.

The overall performance of computational time for each compared method can be evaluated by plots of the average encoding time vs. different QPs and the average decoding time vs. different QPs. The five curves in Fig. 5 show the plots of the average encoding time vs. different QPs for the five compared methods with the High profile. It is clear that the SE method has the least computational time at the encoding side since it only uses five most frequent prediction modes, whereas the proposed and Laroche et al.'s methods require more computational time due to the examination of the redundant prediction modes. Nevertheless, for the cases of higher QPs, the computational time of the proposed and Laroche et al.'s methods may be less than that of H.264/AVC since more redundant prediction modes are eliminated, and hence, the RD optimization for the optimal prediction mode would be accelerated. When compared with Laroche et al.'s method, the proposed method always requires less computational time. The reason is that more similarity checkings are required for eliminating redundant prediction modes by an exhaustive search in Laroche et al.'s method than that by the Hamilton cycle in the proposed method, while the two methods have the similar computational time for the RD optimization, as shown in Tables 5 and 6.

Similarly, the plots of the average decoding time vs. different QPs for the concerned five methods are given in Fig. 6 and show that the SE and Kwon et al.'s methods and H.264/AVC

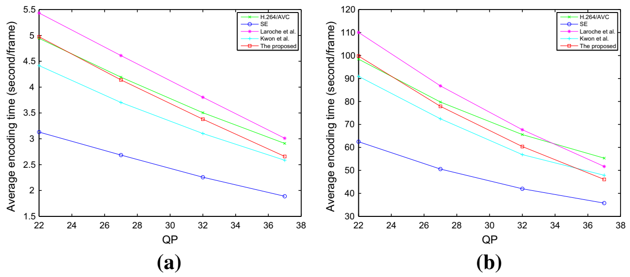

Fig. 5 The average encoding time for each of the five compard methods over different QPs corresponding to the two testing video sequences with the High profile: **a** Tempete (CIF) and **b** CrowdRun (1080p)

Table 5 Average number of the similarity checkings and the corresponding computational time required for the proposed method and Laroche et al.'s method

Method	Block size	Number of similarity checkings (Computational time in μ s)
Laroche et al.'s	4 × 4	25.66 (24.92)
	8 × 8	30.65 (110.24)
Proposed	4 × 4	8.75 (9.62)
	8 × 8	8.81 (34.78)

Table 6 Average number of the remaining prediction modes used in the RD optimization and the corresponding computational time after performing the elimination schemes of the proposed method and Laroche et al.'s method

Method	Block size	Number of remaining prediction modes (Computational time in μ s)
Laroche et al.'s	4 × 4	2.75 (124.42)
	8 × 8	7.38 (374.59)
Proposed	4 × 4	2.90 (124.67)
	8 × 8	7.52 (375.24)

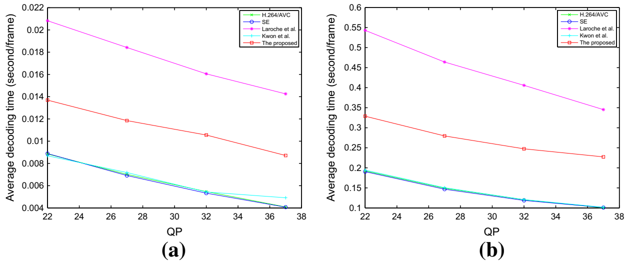

Fig. 6 The average decoding time for each of the five compared methods over different QPs corresponding to the two testing video sequences with the High profile: **a** Tempete (CIF) and **b** CrowdRun (1080p)

have the least and approximately equal computational time at the decoding side since the three methods can directly retrieve the index of the optimal prediction mode. Unlike the above three methods, both the proposed and Laroche et al.'s methods additionally need to reconstruct the index set J^* so as to retrieve the index of the optimal prediction mode, implying that the corresponding elimination schemes would be reperformed, and hence, more computational time is delivered. Nevertheless, when compared with Laroche et al.'s method, the proposed method is still more efficient since when reconstructing the corresponding index sets, the proposed method has less computational time complexity than Laroche et al.'s method.

6 Conclusions

This paper has presented an efficient linear-time elimination method to eliminate the redundant prediction modes of intra-coding in H.264/AVC. When compared with Laroche et al.'s method, the proposed elimination method can substantially reduce the execution time while preserving the similar PSNR and bitrate performance. The proposed elimination method can be extended to the cases when compressing the video sequences using the High Efficiency Video Coding (HEVC) standard [13] and is expected to achieve more significant effect on bitrate reduction. However, it is a challenging to find the Hamilton cycle in the graph with thirty-five nodes.

References

1. Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC,) Joint Video Team of ISO/IEC and ITU-T, 2003
2. Laroche, G., Jung, J., Pesquet-Popescu, B.: Intra coding with prediction mode information inference. IEEE Trans. Circ. Syst. Video Technol **20**(12), 1786–1796 (2011)
3. Chiu, Y.H., Chung, K.L., Lin, C.H.: An improved universal subsampling strategy for compressing mosaic videos with arbitrary RGB color filter arrays in H.264/AVC. J Visual Commun. Image Represent. **25**(7), 1791–1799 (2014)
4. Robert, A., Amonou, I., Pesquet-Popescu, B.: Improving H.264 video coding through block oriented transforms, In: Proc. of 2008 IEEE International Conf. on Multimedia and Expo, Hannover, Germany, 2008, pp. 705–708
5. Ji, X., Kwong, S., Zhao, D., Wang, H., Kou, C.C.J., Dai, Q.: Early determination of zero-quantized 8 × 8 DCT coefficients. IEEE Trans. Circ. Syst. Video Technol. **19**(12), 1755–1765 (2009)
6. Tan, T.K., Sullivan, G., Wedi, T.: "Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 4," ITU-T SG16 Q6 (VCEG) document VCEG-AJ10r1 (2008)
7. Gross, J.L., Yellen, J.: Graph Theory and Its Applications, 2nd edn. Chapman and Hall/CRC, London (2005)
8. [Online]. Available: http://media.xiph.org/video/derf/
9. [Online]. Available: http://www.lvideoservices.net/hd_links.htm
10. Kwon, S.K., Punchihewa, A., Bailey, D.G., Kim, S.W., Lee, J.: Adaptive Simplification of Prediction Modes for H.264 Intra-Picture Coding. IEEE Trans. Broadcasting **58**(1), 125–129 (2012)
11. [Online]. Available: http://iphome.hhi.de/suehring/tml/download/jm18.5.zip
12. Bjøntegaard, G.: "Calculation of average PSNR differences between RD curves," VCEG contribution VCEG-M33, Austin, (2001)
13. Sullivan, G.J., Ohm, J., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circ. Syst. Video Technol. **22**(12), 1649–1668 (2012)