An Efficient Intra Prediction Algorithm for H.264/AVC High Profile

Bo Shen[1] Kuo-Hsiang Cheng[2] Yun Liu[1] Ying-Hong Wang[2*]

[1] School of Electronic and Information Engineering, Beijing Jiaotong University
Beijing, 100044, China
[2]Department of Computer Science and Information Engineering, Tamkang University,
New Taipei City 25137, Taiwan, ROC
bshen@bjtu.edu.cn, alex111@ms6.hinet.net, liuyun@bjtu.edu.cn, inhon@mail.tku.edu.tw

Received 23 August 2013; Revised 12 December 2013; Accepted 16 December 2013

Abstract. A simple, highly efficient intra prediction algorithm to reduce the computational complexity of H.264/AVC High Profile is proposed. The algorithm combines two methods. The first method is a quant-based block-size selection decision that is based on the sum of the quantization AC coefficients among intra 8 × 8 mode predictions, combined with an error adjustment to select either intra 4 × 4 or intra 16 × 16 mode predictions. The second method is a novel direction-based prediction mode decision that is used to reduce the possible prediction modes for the rate-distortion (RD) optimization technique. Our experimental results demonstrate that the proposed algorithm reduces the encoding time by approximately 54% compared with that needed for an exhaustive search using the joint model reference software. The peak signal-to-noise ratio degradation is negligible, and the bit rate increment is minimal. Furthermore, the results show that our algorithm achieves a significant improvement in both computation performance and RD performance as compared with the existing algorithms.

Keywords: H.264/AVC, block size, prediction mode, rate-distortion optimization

1 Introduction

The H.264/Advanced Video Coding (AVC) standard [1], also known as MPEG-4 Part-10 AVC, was developed by the Joint Video Team (JVT). JVT is a group of video coding experts from the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). This standard significantly improves the coding performance, yielding a shorter bit rate and operating at the same peak signal-to-noise ratio (PSNR) as the previous video coding standards. As compared with MPEG-4, H.263, and MPEG-2, H.264/AVC can save up to 39%, 49%, and 64% of the bit rate, respectively [2].

The H.264/AVC standard adopts a number of advanced coding tools, such as variable block sizes, multiple reference frames, multiple intra prediction modes, rate-distortion optimization (RDO), and exhaustive search decision [2]. In inter mode coding, a macroblock (MB) can be divided into 16 × 16, 16 × 8, 8 × 16, 8 × 8, 8 × 4, 4 × 8, and 4 × 4 blocks for motion estimation. In intra mode coding, an MB can be divided into intra 16 × 16 (I16 MB), intra 8 × 8 (I8MB, High Profile only [2-3]), and intra 4 × 4 (I4MB) blocks for intra prediction. In intra-mode coding, there are nine prediction modes for I4MB and I8MB and four prediction modes for I16MB. In the RDO technique, H.264/AVC defines the rate-distortion (RD) cost function as follows [4-5]:

$$J(s,c,MODE|QP) = SSD(s,c,MODE|QP) + \lambda_{MODE} \cdot R(s,c,MODE|QP), \tag{1}$$

where *QP* is the quantization parameter; *SSD* denotes the sum of the squared difference between *s* and *c,* which is the distortion metric; *s* and *c* indicate the original block and its reconstructed block, respectively; *MODE* denotes an MB mode; λ is a Lagrange multiplier that weighs the influence of both distortion and bit rate in the RD cost function; and *R* represents the number of bits associated with the currently selected MB mode *MODE*.

To achieve the best coding performance, the RDO technique compares all possible combinations of modes to find the minimum RD cost. Unfortunately, this exhaustive search process results in drastic computational complexity [6-7]. Therefore, the development of fast algorithms that reduce computational complexity and preserve video quality and bit rate, as close as possible to those of an exhaustive search, has become the primary factor for practical H.264/AVC coding.

In general, the coding performance of inter mode coding is superior to intra mode coding because inter mode coding offers a shorter bit rate. However, intra mode coding is essential for scene changes, light condition changes, the appearance of a new object, and error recovery. A number of algorithms have been previously presented to reduce the computational complexity of intra mode coding. These algorithms can be classified into two categories: block-size selection decision and fast prediction mode decision.

For block-size selection decision, a smaller block size prediction mode (I4MB) is generally more suitable for MBs with complex texture, while a larger block size prediction mode (I16MB) is more suitable for smooth MBs. For example, Yu *et al.* [8] and Huang *et al.* [9] used the variance classification of texture complexity to filter out I4MB or I16MB mode prediction. Wei *et al.* presented a method that uses the variance of DC coefficients to eliminate the I4MB or I16MB prediction mode [10]. Lin *et al.* [11] used the sum of absolute low-frequency components of a discrete cosine transform (DCT) block of an MB as a flatness degree, which is used to select the I4MB or I16MB prediction mode. In another study, Lim *et al.* [12] used the variance of an inner-8 × 8 sub-block of an MB as a threshold to select either the I4MB or I16MB prediction mode. Zhang *et al.* [27] presented an algorithm that using DC ratio to select only one or two block types in 4 × 4, 8 × 8 and 16 × 16 intra prediction instead of three of them.

For fast prediction mode decision, many fast intra prediction mode methods have been proposed. To reduce computational complexity, these methods can be divided into two types. One type modifies the match criterion to reduce the RD cost. For example, in [13], Tseng *et al.* proposed an enhanced RD function that combines the sum of absolute integer-transformed differences (SAITD) as a distortion and a rate predictor to obtain the approximate rate for I4MB prediction. Lee *et al.* [14] proposed a mode decision algorithm based on the sum of absolute transformed differences (SATD) that incorporates both SATD and the variance information into the RDO mode decision method, and presented an improved algorithm for their own algorithm [11].

The other type employs edge direction detection to select a small part of prediction modes for RDO calculations. For example, Pan *et al.* [15] proposed a fast intra prediction algorithm that extracts image features using Sobel edge operators and selects the predictor according to their statistics. Wang *et al.* exploited the features of the edge histogram descriptors used in MPEG-7 to detect the dominant edge direction in order to reduce the possible prediction modes [16]. Tsai *et al.* [17] calculated the gradient of four selected pixels of a block to extract the orientation of the block.

Furthermore, Tsai *et al.* [18] proposed two direction detection algorithms by computing sub-block and pixel direction differences. Kim *et al.* presented an adaptive prediction method that is based on the variance of the block boundary pixels [19]. Quan *et al.* [20] presented an algorithm that used the ratio of the variance along the horizontal direction to that along the vertical direction to select a small part of prediction modes. Adibelli *et al.* [21] proposed a technique that performs a small number of comparisons among the current block's neighboring pixels prior to the intra prediction process. Zeng *et al.* [22] proposed an algorithm to reduce the number of prediction modes according to their Hadamard distances and prediction directions. Huang *et al.* [9] presented an algorithm based on [16] by adding the most probable mode (MPM) to the algorithm. Lim *et al.* [12] used the similarity of the reference pixels to restrict the search of the prediction modes.

In this study, two additional efficient methods that reduce the computational complexity of H.264/AVC High Profile are proposed. The first method is a quant-based block-size selection decision and the second is a novel direction-based prediction mode decision. These methods form a highly efficient intra prediction algorithm.

The rest of this paper is organized as follows. In Section 2, intra mode coding in H.264/AVC is reviewed. The proposed algorithms are described in Section 3. Section 4 presents and compares the experimental results, and Section 5 presents the conclusions.

2 H.264/AVC Intra Prediction Overview

The intra prediction algorithm predicts the pixels in an MB using the pixels in the available neighboring blocks. For the luminance (luma) component of an MB, a 16 × 16 predicted luma block is formed by performing intra prediction for each 4 × 4 luma block in the MB and for the 16 × 16 MB. There are nine prediction modes for each 4 × 4 luma block and four prediction modes for a 16 × 16 luma block. A mode decision algorithm is then used to compare the 4 × 4 and 16 × 16 predictions and to select the best luma prediction mode for the MB. In general, 4 × 4 prediction modes are selected for highly textured regions, while 16 × 16 prediction modes are selected for flat regions.

Nine 4 × 4 luma prediction modes are designed in a directional manner. A 4 × 4 luma block consisting of pixels a–p is shown in Fig. 1. The pixels A–M belong to the neighboring blocks and are assumed to be already encoded and reconstructed; therefore, they are available in the encoder and decoder to generate a prediction for the current MB. Each 4 × 4 luma prediction mode generates 16 predicted pixel values using some or all of the neighboring pixels A–M, as shown in Fig. 2. The arrows indicate the directions of prediction in each mode. The predicted pixels are calculated by a weighted average of the neighboring pixels A–M for each mode, except mode2 (DC) which is the average of the pixels A-M. In the I16MB prediction block, a uniform prediction is performed for all luminance components of an MB using four prediction modes, as shown in Fig. 3.

The intra prediction mode for the 8 × 8 luma block has been adopted in H.264/AVC High Profile and is known as the fidelity range extensions (FRExt) [23]. The I8MB prediction modes are the same as the I4MB prediction modes, except that the block size of I8MB is an 8 × 8 matrix. Because the high profile is primarily targeted at providing significant improvements in the coding efficiency for higher-fidelity video material [24] and our target is high-resolution video sequences, we focus on the high profile in this study.

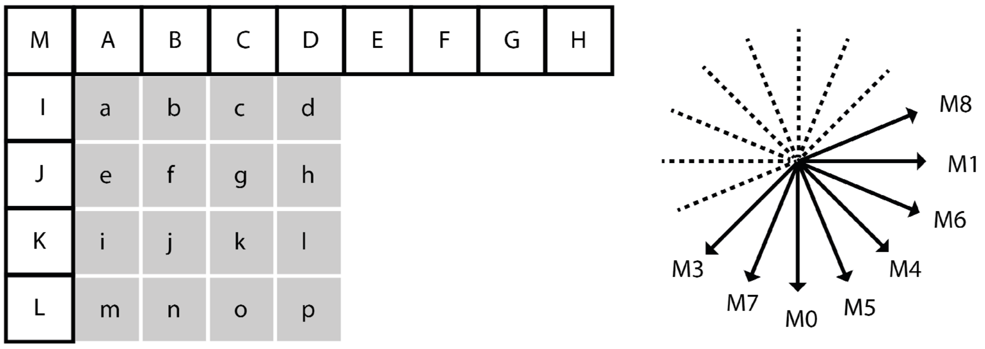


Fig. 1. Labelling of prediction samples (4 × 4)

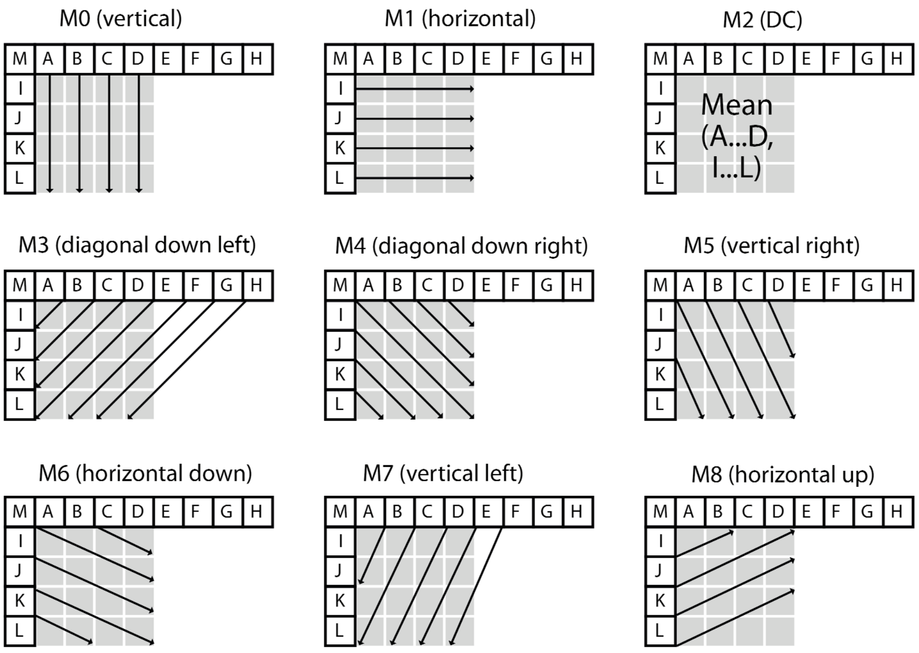


Fig. 2. Intra 4 × 4 luma prediction modes

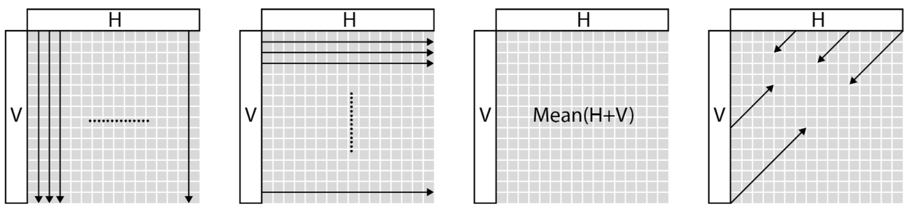


Fig. 3. Intra 16 × 16 luma prediction mode

3 Proposed Algorithm

As shown in Fig. 4, in the H.264/AVC normal procedure, an intra frame performs intra prediction to select the best intra mode, and an inter frame performs intra prediction after motion estimation. Figure 5 shows a flowchart of our

H.264/AVC compliant intra encoder. The blue boxes in Fig. 5 form a quant-based block-size selection decision to filter out I4MB prediction or I16MB prediction depending on the pre-calculated information obtained from I8MB prediction. The two red boxes are fast intra mode prediction decisions for I4MB and I8MB predictions. Because I16MB prediction has only four prediction modes and is significantly faster than I4MB and I8MB predictions, therefore, we remain I16MB prediction to the normal process.

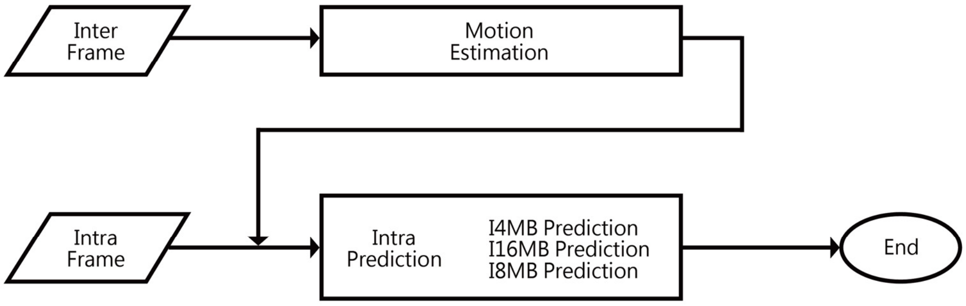


Fig. 4. H.264 encoder flowchart

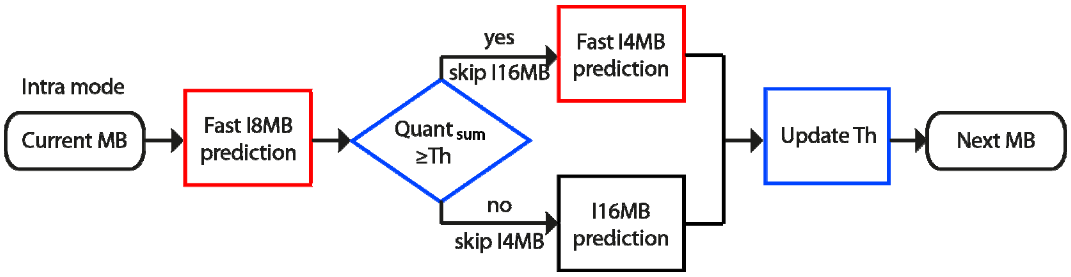


Fig. 5. H.264 compliant intra encoder flowchart

3.1 Quant-based block-size selection decision

In H.264/AVC High Profile, the additional integer 8 × 8 DCT is used for I8MB prediction. As shown in Fig. 6, after DCT, the upper left element is referred to as the DC coefficient, which represents the average of the original I8MB prediction. All other elements are referred to as the AC coefficient, which represents the variance across the 8 × 8 block. The AC coefficient provides a very good method for classifying texture images [9, 14]; the more complex the texture, the larger the AC coefficient.

In our algorithm, we exploit the AC coefficient characteristics for block size selection, represented by blue boxes in Fig. 5. First, the I8MB prediction is processed. Then, the sum of the absolute AC coefficients from the I8MB prediction is used to determine either the I4MB or I16MB prediction. In H.264/AVC, multiplication scaling (part of the DCT) is integrated into the quantization process, as shown in Fig. 7. The sum of absolute AC coefficients in the middle of the quantizer from four optimal I8MBs is summarized as follows:

$$\mathrm{Qaunt}_{\mathrm{ACsum}} = \sum_{i=0}^{3}\sum_{j=1}^{63}(\text{absolute quant coefficient}). \tag{2}$$

This has two advantages: (1) No time-consuming operations are needed to calculate $\mathrm{Quant}_{\mathrm{ACsum}}$; the only operation required is to summarize the absolute AC coefficients, and (2) $\mathrm{Quant}_{\mathrm{ACsum}}$ considers the QP; that is applicable to a variety of different values of QP.

As the first step, we assume that $\mathrm{Quant}_{\mathrm{ACsum}}$ of all I4MB, I8MB, and I16MB are normal distributions, as shown in Fig. 8. In the figure, the average of $\mathrm{Quant}_{\mathrm{ACsum}}$ is assumed to be a perfect adaptive threshold to select I4MB or I16MB prediction. If $\mathrm{Quant}_{\mathrm{ACsum}}$ is greater than the threshold, I4MB prediction is selected; otherwise, I16MB prediction is selected. We initialize the threshold to $\mathrm{Quant}_{\mathrm{ACsum}}$ after accomplishing the first I8MB prediction. Then, when either I4MB or I16MB prediction is completed, we update the threshold to the average of $\mathrm{Quant}_{\mathrm{ACsum}}$ if I8MB prediction is the optimal mode. Our first algorithm to update the threshold is summarized in Algorithm 1.

As the first step, we assume that $\mathrm{Quant}_{\mathrm{ACsum}}$ of all I4MB, I8MB, and I16MB are normal distributions, as shown in Fig. 8. In the figure, the average of $\mathrm{Quant}_{\mathrm{ACsum}}$ is assumed to be a perfect adaptive threshold to select I4MB or I16MB prediction. If $\mathrm{Quant}_{\mathrm{ACsum}}$ is greater than the threshold, I4MB prediction is selected; otherwise, I16MB prediction is selected. We initialize the threshold to $\mathrm{Quant}_{\mathrm{ACsum}}$ after accomplishing the first I8MB prediction. Then, when either I4MB or I16MB prediction is completed, we update the threshold to the average of $\mathrm{Quant}_{\mathrm{ACsum}}$ if I8MB prediction is the optimal mode.

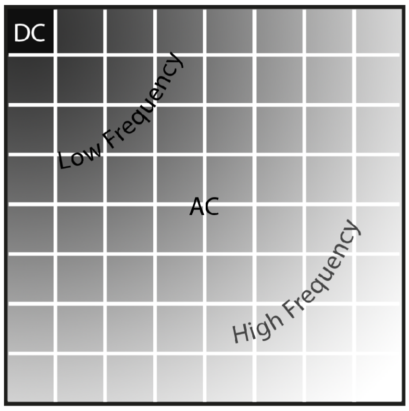


Fig. 6. Integer 8 × 8 DCT

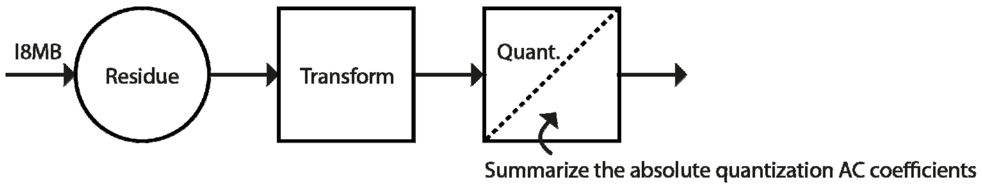


Fig. 7. Summarize the absolute quantization AC coefficients

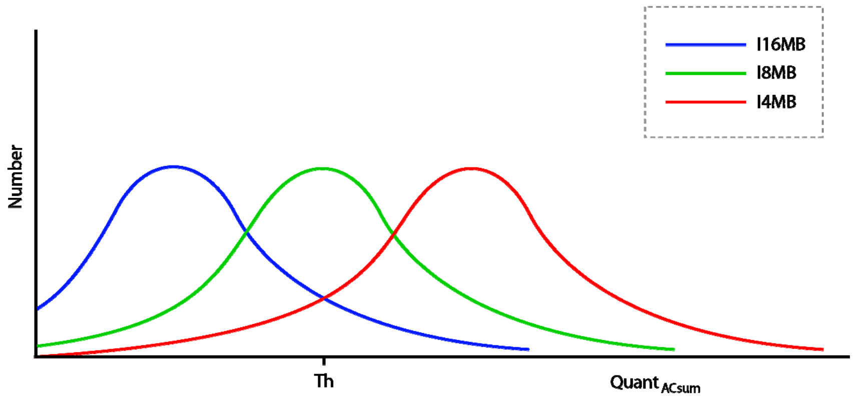


Fig. 8. Distribution of QuantACsum for I4MB, I8MB, and I16MB

Line 1 in Algorithm 1, total_number is initialized to 0. In line 2 (indicate the first red box in Fig. 5), the RD cost (I8MB_rdcost) and best mode (I8MB_best_mode) for I8MB can be obtained in the normal I8MB prediction process, the extra computation is to calculate $\text{Quant}_{\text{ACsum}}$. Line 3 indicates the diamond box in Fig. 5, the threshold (Th) is initialized to $\text{Quant}_{\text{ACsum}}$ when first run. Line 4-9 perform the I4MB prediction process that has shown in the second red box in the Fig 5, I4MB_rdcost and I4MB_best_mode denote the RD cost and the best mode for I4MB, respectively. Similarly, Line 11-16 perform the I16MB prediction process (the black box in the Fig 5), I16MB_rdcost and I16MB_best_mode denote the RD cost and the best mode for I16MB, respectively. Finally, Line 18-19 is to update the threshold (the blue box in the Fig 5), and Quant_total is initial to 0.

We applied Algorithm 1 (which updates the threshold for block size selection) to the JVT reference software (version JM13.2) [25] for four test sequences with a QP ranging from 16 to 32 to check the hit rate (HR), which was found to be comparable with that obtained by the variance-based algorithm [9]. The HR indicates that the accuracy ratio of the best mode selected using the block size selection algorithm is the same as that of the best mode selected using an exhaustive search by the JM13.2. The results are shown in Table 1. However, HR performance using Algorithm 1 is only 86.38% on average, which is worse than that using the variance-based algorithm (90.44% on average). We determined that there were two types of errors in the distribution assumed in Fig. 8. We refer to these error types as an unbalance error and an I8MB curve bias error.

The following is an example of an unbalance error. In the mobile test sequence (QP = 28), optimal I4MB is 77.26%; however, optimal I16MB only occupies 3%. The distribution in this situation is illustrated in Fig. 9. It is evident that the threshold should be moved to the left when the amount of optimal I4MB is larger than that of optimal I16MB. Therefore, we adjusted the threshold relative to the amount of optimal I4MB and I16MB, i.e., if the amount of optimal I4MB is larger than that of optimal I16MB, then the threshold is decreased; otherwise, the threshold is increased. Our second algorithm to update the threshold is summarized in Algorithm 2.

Algorithm 1: Update the threshold for block size selection

```
1:  ++total_number
2:  perform I8MB prediction process and calculate $Quant_{ACsum}$
3:  if $Quant_{ACsum} \geq Th$ then
4:      perform I4MB prediction process
5:      if I8MB_rdcost < I4MB_rdcost then
6:          best mode = I8MB_best_mode
7:      else
8:          best mode = I4MB_best_mode
9:      end if
10: else
11:     perform I16MB prediction process
12:     if I8MB_rdcost < I16MB_rdcost then
13:         best mode = I8MB_best_mode
14:     else
15:         best mode = I16MB_best_mode
16:     end if
17: end if
18: Quant_total += $Quant_{ACsum}$
19: $Th = \frac{Quant\_total}{total\_number}$
```

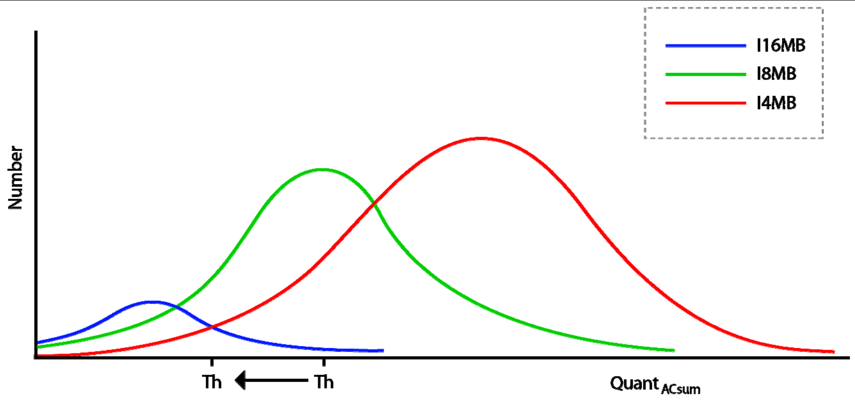


Fig. 9. Distribution of QuantACsum for I4MB, I8MB, and I16MB (unbalance error)

I4MB_number, I8MB_number and I16MB_number, in Algorithm 2, are all initialized to 0. Note that the ratio in line 24, if the ratio is greater than 1 that means the amount of optimal I4MB is larger than that of optimal I16MB, so that the threshold is decreased (line 25); otherwise, the threshold is increased.

As shown in Table 1, the HR produced by Algorithm 2 is superior to the HR produced by Algorithm 1. On average, the HR improved from 86.38% up to 94.16%. In addition, the HR accuracy of Algorithm 2 (94.16%) is much better than that of the variance-based algorithm (90.44%).

The second error type, an I8MB curve bias error, occurs when the distribution curve of I8MB is not in the middle of the I4MB and I16MB distribution curves. For example, as shown in Fig. 10, the I8MB distribution curve is biased toward the I4MB distribution curve. In this situation, the threshold should slightly shift to left. To recover the I8MB curve bias error, we analyzed four conditions, which are shown in Table 2. For Condition 1, the percentage of the I8MB_rdcost (I8MB RD cost) is less than that of the I16MB RD cost (I16MB RD cost) when the optimal mode is I4MB. The values are obtained by performing exhaustive search. Conditions 2–4 have the similar meaning like Condition 1.

It is evident that the likelihood of Condition 1 is much greater than that of Condition 2. Therefore, we can deduce that when I16MB prediction is selected (i.e., skip I4MB prediction) and I8MB_rdcost is less than I16MB_rdcost (Condition 1), in this situation, the error possibility of the selection is greater than that when I16MB prediction is selected and I8MB_rdcost is greater than I16MB_rdcost (Condition 2). So , we can assume that when I16MB prediction is selected and I8MB_rdcost is less than I16MB_rdcost, the selection is supposed to be incorrect, and decrease the threshold (shift the threshold to the left that is shown as Fig. 9) that let the I4MB selection made greater advantage when process next MB; otherwise, if I8MB_rdcost is greater than I16MB_rdcost , we assume that the I16MB selection is correct and increase the threshold. Conditions 3 and 4 for I4MB selection can be deduced in a similar manner by skipping_I16MB prediction. Our final algorithm to update the threshold is summarized in Algorithm 3.

Algorithm 2: Update the threshold for block size selection

```
1:  total_number++
2:  perform I8MB prediction process and calculate $Quant_{ACsum}$
3:  if $Quant_{ACsum} \geq Th$ then
4:      perform I4MB prediction process
5:      if I8MB_rdcost < I4MB_rdcost then
6:          best mode = I8MB_best_mode
7:          Quant_total += $Quant_{ACsum}$
8:          Quant_avg = $\frac{Quant\_total}{++I8MB\_number}$
9:      else
10:         best mode = I4MB_best_mode
11:         I4MB_number++
12:     end if
13: else
14:     perform I16MB prediction process
15:     if I8MB_rdcost < I16MB_rdcost then
16:         best mode = I8MB_best_mode
17:         Quant_total += $Quant_{ACsum}$
18:         Quant_avg = $\frac{Quant\_total}{++I8MB\_number}$
19:     else
20:         best mode = I16MB_best_mode
21:         I16MB_number++
22:     end if
23: end if
24: ratio += $1.0 - \frac{I4\_number - I16\_number}{total\_number}$
25: Th = Quant_avg × ratio
```

Table 1. Hit rate for block size selection algorithms

Test Sequence	QP	Variance-based algorithm [9]	Algorithm 1	Algorithm 2	Algorithm 3
Mobile (CIF)	16	92.82	84.60	99.01	99.10
	20	93.24	77.78	97.73	99.04
	24	93.18	73.68	96.20	98.53
	28	93.98	73.59	94.65	99.02
	32	94.59	72.65	92.08	98.34
Average		93.56	74.46	95.93	98.80
Container (CIF)	16	92.95	94.48	97.54	97.86
	20	92.26	94.59	95.38	97.02
	24	91.08	93.85	94.57	95.53
	28	90.20	92.16	92.10	91.99
	32	89.90	92.04	91.74	91.23
Average		91.28	93.42	94.27	94.73
Parkrun (720p)	16	82.70	78.40	94.92	99.49
	20	93.80	89.50	94.34	96.93
	24	93.20	88.93	92.03	94.60
	28	96.52	92.74	93.92	95.46
	32	97.92	92.98	94.01	95.13
Average		92.83	88.51	93.84	96.32
Shields (720p)	16	75.70	84.72	96.23	98.04
	20	84.16	89.52	95.78	96.27
	24	83.80	87.03	93.11	96.47
	28	86.45	86.53	88.97	93.22
	32	90.29	87.88	88.84	91.54
Average		84.04	87.13	92.59	95.11
Total Average		90.44	86.38	94.16	95.76

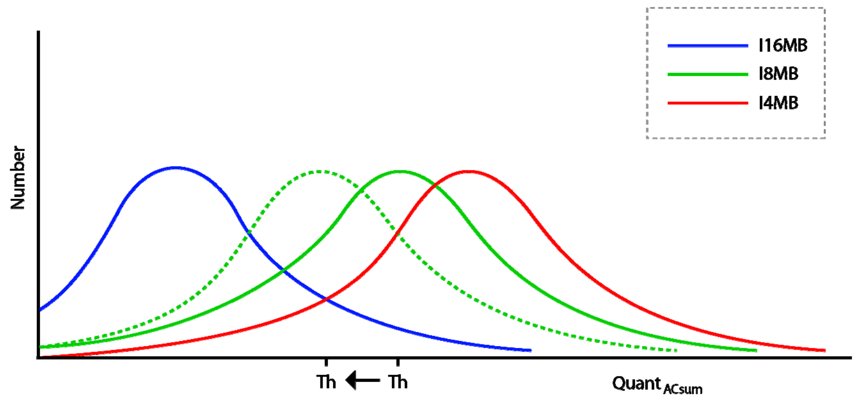


Fig. 10. Distribution of QuantACsum for I4MB, I8MB, and I16MB (I8MB curve bias error)

Algorithm 3: Update the threshold for block-size selection

```
Line 1–23 are identical to Algorithm 2
24:  if I4MB_number > 0 and I8MB_number > 0 and I16MB_number > 0 then
25:      ratio += $1.0 - \frac{\text{I4MB\_number} - \text{I16MB\_number}}{\text{total\_number}}$
26:      ratio_left_bound = ratio − 0.2
27:      ratio_right_bound = ratio + 0.2
28:      if Quant$_{ACsum}$ $\geq$ Th then                         // I4MB prediction selected
29:          if I8MB_rdcost < I4MB_rdcost then                     // Condition 3
30:              if ratio < ratio_right_bound then
31:                  ratio += $0.1 \times \frac{\text{I4MB\_rdcost} - \text{I8MB\_rdcost}}{\text{I4MB\_rdcost} + \text{I8MB\_rdcost}}$
32:              else
33:                  ratio = ratio_right_bound
34:              end if
35:          else                                                  // Condition 4
36:              if ratio > ratio_left_bound then
37:                  ratio −= $0.1 \times \frac{\text{I8MB\_rdcost} - \text{I4MB\_rdcost}}{\text{I4MB\_rdcost} + \text{I8MB\_rdcost}}$
38:              else
39:                  ratio = ratio _left_bound
40:              end if
41:      else                                                      // I8MB prediction selected
42:          if I8MB_rdcost < I16MB_rdcost then                    // Condition 1
43:              if ratio > ratio_left_bound then
44:                  ratio −= $0.1 \times \frac{\text{I16MB\_rdcost} - \text{I8MB\_rdcost}}{\text{I16MB\_rdcost} + \text{I8MB\_rdcost}}$
45:              else
46:                  ratio = ratio _left_bound
47:              end if
48:          else                                                  // Condition 2
49:              if ratio > ratio_right_bound then
50:                  ratio += $0.1 \times \frac{\text{I8MB\_rdcost} - \text{I16MB\_rdcost}}{\text{I16MB\_rdcost} + \text{I8MB\_rdcost}}$
51:              else
52:                  ratio = ratio_right_bound
53:          end if
54:      end if
55:  end if
56:  Th = Quant_avg × ratio
```

In Algorithm 3, lines 1–23 are identical to Algorithm 2. Lines 25–27 calculate the left and right ratio boundaries, which keep the ratio inside the boundary to avoid the worst case, and in our enormous experiments, the left ratio boundary minus 0.2 and right boundary plus 0.2 that to make the ratio boundary wider can get the best results. The value Line 31 and line 50 increase the ratio, line 37 and line 44 decrease the ratio when I4MB prediction is skipped, as described in the previous paragraph, and among those lines, the value 0.1 is to make ratio increases or decreases a small step. As shown in Table 1, the HR using Algorithm 3 is better than that using Algorithm 2; on average, it has improved from 94.16% up to 95.76%.

Table 2. Percentages for four conditions

Test Sequence	Optimal best mode = I4MB		Optimal best mode = I16MB	
	Condition 1: I8MB_rdcost < I16MB_rdcost (%)	Condition 2: I8MB_rdcost ≥ I16MB_rdcost (%)	Condition 3: I8MB_rdcost < I4MB_rdcost (%)	Condition 4: I8MB_rdcost ≥ I4MB_rdcost (%)
Mobile (CIF)	70.25	11.48	1.53	1.46
Container (CIF)	31.96	4.67	9.33	5.21
Parkrun (720p)	35.88	1.17	1.65	0.32
Shields (720p)	41.19	2.29	12.17	1.69

3.2 Direction-based prediction mode decision

Most edge direction detection methods use edge detectors to determine dominant directions and to reduce the number of prediction modes. However, most methods do not consider information from neighboring blocks. We improve the traditional edge direction detection methods by including related neighboring blocks.

First, we calculate three variances: σ_{VER} (variance of the vertical mode), σ_{HOR} (variance of the horizontal mode), and σ_{DDR} (variance of the diagonal down-right mode) when the upper left blocks are available, as shown in Figs. 11 (a), (b), and (c), respectively. The fourth variance, σ_{DDL} (variance of the down-left mode), can be calculated when the upper right block is also available, as shown in Fig. 11 (d). Equations (3), (4), (5) and (6) are used to, respectively, calculate σ_{VER}, σ_{HOR}, σ_{DDR} And σ_{DDL} as follows.

$$\sigma_{VER} = \sum_{x=1}^{4}\sum_{y=1}^{4}\{I(x,y) - I(x,0)\}^2 \tag{3}$$

$$\sigma_{HOR} = \sum_{y=1}^{4}\sum_{x=1}^{4}\{I(x,y) - I(0,y)\}^2 \tag{4}$$

$$\sigma_{DDR} = \sum_{x=1}^{4}\sum_{y=1}^{x}\{I(x,y) - I(x-y,0)\}^2 + \sum_{x=1}^{3}\sum_{y=x+1}^{4}\{I(x,y) - I(0,y-x)\}^2 \tag{5}$$

$$\sigma_{DDL} = \sum_{y=1}^{4}\sum_{x=1}^{4}\{I(x,y) - I(x+y,0)\}^2 \tag{6}$$

where *x* and *y* denote the position of the pixel; The pixels *I*(*x*, *y*) belong to the neighboring blocks those are already encoded and reconstructed when *x* or *y* equals to 0; The pixels *I*(*x*, *y*) belong to the luma 4 × 4 block when both *x* and *y* are greater than 0.

Next, the ratio of the smallest and the second smallest mode from the variance set {σ_{VER}, σ_{HOR}, σ_{DDR} [σ_{DDL}]} is calculated. If the ratio (smallest variance / second smallest variance) is less than the threshold, a direction exists. Then, the mode with the smallest variance and its two neighboring modes plus the MPM, or the DC mode when MPM is the same as the first three modes, are selected as candidate modes. The MPM is defined as the prediction mode of the left or upper neighbor, whichever has the smaller prediction number. The MPM is the only candidate mode when σ_{VER}, σ_{HOR}, and σ_{DDR} are equal. When σ_{VER}, σ_{HOR}, and σ_{DDR} are equal, there is a high probability (approximately 90% in our experiment) that the samples in the prediction reference (labelled A–M in Fig. 1) have the same value. Under this condition, the prediction values of the nine prediction modes are identical; therefore, only one prediction mode is required. In our algorithm, we select the MPM as the single candidate mode because in the H.264 syntax, only one bit flag is needed to signal MPM, while four bits are needed to signal a non-MPM. Selecting the MPM as the single candidate mode can save three bits for one block coding.

Finally, we propose our fast intra prediction mode algorithm for I4MB in Algorithm 4.

The symbols: up_block_available, left_block_available and up_right_block_available in Algorithm 4, indicate that if the preceding block has up, left or upright encoded block. And, M0, M1, M3, M4, M5, M6, M7, M8 and DC are indicated in Fig. 2.

An 8 × 8 block of the I8MB mode has nine prediction modes that are identical to the I4MB mode, except for the block size. Consequently, the I8MB fast prediction mode decision is almost identical to the I4MB fast prediction mode decision, but the threshold (Th_{I8}) can be different from Th_{I4}. We applied this algorithm to JM13.2 with a threshold ranging from 0.9 to 1.0 in order to check the HR and the filter rate (FR). The FR is the percentage of blocks selected to filter out improper prediction modes. In the results of our algorithm, shown in Table 3, the threshold (Th) was set to Th_{I4} and Th_{I8}. A smaller threshold produces a higher HR, improving the coding performance; however, a higher HR decreases the FR, increasing the computation time.

Algorithm 4: Direction-based prediction mode decision

```
1:   if up_block_available and left_block_available then
2:       calculate $\sigma_{VER}$ and $\sigma_{HOR}$
3:       if $\sigma_{VER} \leq \sigma_{HOR}$ then
4:           smallest =$\sigma_{VER}$
5:           secondsmallest = $\sigma_{HOR}$
6:           probable_best_mode = M0
7:       else
8:           smallest = $\sigma_{HOR}$
9:           secondsmallest =$\sigma_{VER}$
10:          probable_best_mode = M1
11:      end if
12:      calculate $\sigma_{DDR}$
13:      if $\sigma_{DDR}$ < smallest then
14:          secondsmallest = smallest
15:          smallest = $\sigma_{DDR}$
16:          probable_best_mode = M4
17:      else
18:          if $\sigma_{DDR}$ < secondsmall then
19:              secondsmallest = $\sigma_{DDR}$
20:          end if
21:      end if
22:      if up_right_block_available then
23:          calculate $\sigma_{DDL}$
24:          if $\sigma_{DDL}$ < smallest then
25:              secondsmallest = smallest
26:              smallest = $\sigma_{DDL}$
27:              probable best mode = M3
28:      else
29:          if $\sigma_{DDL}$ < secondsmallest then
30:              secondsmallest = $\sigma_{DDL}$
31:          end if
32:      end if
33:  end if
34:
35:  ratio = smallest / secondsmall
36:  if ratio < $Th_{I4}$ then
37:      if probable_best_mode equals M0 then
38:          candidate mode set = {M0, M5, M7, MPM or DC}
39:      end if
40:      if probable_best_mode equals M1 then
41:          candidate mode set = {M1, M6, M8, MPM or DC}
42:      end if
43:      if probable_best_mode equals M3 then
44:          candidate mode set = {M3, M7, M8, MPM or DC}
45:      end if
46:      if probable_best_mode equals M4 then
47:          candidate mode set = {M4, M5, M6, MPM or DC}
48:      end if
49:  else
50:      if $\sigma_{VER}$, $\sigma_{HOR}$ and $\sigma_{DDR}$ are equal then
51:          candidate mode set = { MPM }
52:      else
53           candidate mode set = {all modes}
54:      end if
55:  end if
```

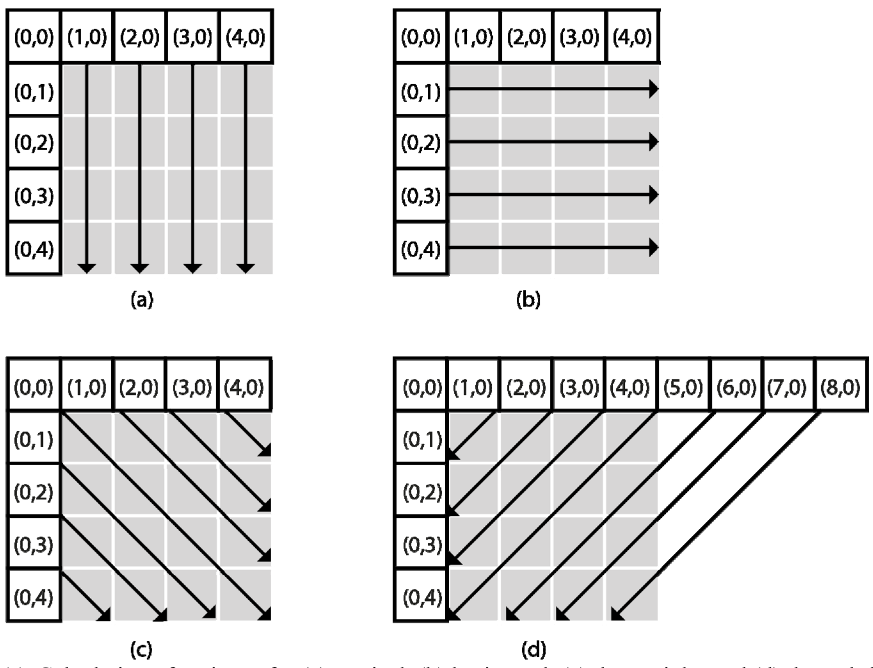


Fig. 11. Calculation of variance for (a) vertical, (b) horizontal, (c) down-right, and (d) down-left modes

Table 3. Hit rate and filter rate for fast intra prediction mode algorithm

Test Sequence	Th	I8MB FR (%)	I8MB HR (%)	I4MB FR(%)	I4MB HR (%)
Mobile (CIF)	1.0	94.43	76.43	95.56	82.01
	0.95	82.13	81.38	88.40	84.52
	0.9	72.87	84.90	81.93	86.67
Container (CIF)	1.0	93.09	84.96	93.73	87.82
	0.95	83.38	87.86	87.65	89.41
	0.9	74.78	90.24	81.42	90.91
Parkrun (720p)	1.0	98.10	74.40	98.73	75.75
	0.95	87.96	78.60	91.47	78.58
	0.9	78.33	82.39	84.28	81.27
Shields (720p)	1.0	97.62	80.11	97.17	82.18
	0.95	88.51	83.33	90.27	84.30
	0.9	80.08	86.20	83.29	86.35

4 Experiments

To verify the performance of our proposed algorithms, we conducted several simulations on various test sequences after implementing the proposed quant-based block-size selection decision algorithm and direction-based prediction mode decision algorithm using the JM13.2. Because most current methods, except [9], do not involve the high profile or simultaneously cover both block-size decision and prediction mode decision, we adopted the same parameter settings used in the study by Huang et al. [9]. All test sequences were performed with RDO enabled. The intra period was set to 1 for intra frame coding, and 15 for inter frame coding; Th_{I8} was 0.9 and Th_{I4} was 0.95 in the direction-based prediction mode decision. Table 4 shows the results of the simulation that applied two QCIF-sized, four CIF-sized, two 720p-sized, and four full HD format test sequences. In Table 4, we used Bjontegaard delta PSNR (BDPSNR) and Bjontegaard delta bit rate (BDBR) [26] as the measuring tools and defined $\Delta PSNR$, ΔBR, and the time saving factor ΔTS as follows:

$$\Delta PSNR = PSNR_{pro} - PSNR_{ref} \quad \textbf{(7)}$$

$$\Delta BR = \frac{BR_{pro} - BR_{ref}}{BR_{ref}} \times 100\,(\%) \quad \textbf{(8)}$$

$$\Delta TS = \frac{T_{ref}-T_{pro}}{T_{ref}} \times 100(\%), \tag{9}$$

Where $PSNR_{pro}$, BR_{pro}, and T_{pro} denote the PSNR of the proposed algorithm, the bit rate of the proposed algorithm, and the time consumed by the proposed algorithm, respectively. $PSNR_{ref}$, BR_{ref}, and T_{ref} denote the PSNR of the reference software, the bit rate of the reference software, and the time consumed by the reference software, respectively.

As shown in Table 4, when the quant-based block-size decision is used independently, the PSNR degraded by 0.002 dB, and the bit rate increased by 0.11% on average; both results are negligible. The average time savings increased by 22.39% when compared with an exhaustive search using the JM13.2. When the direction-based prediction mode decision is independently employed, the PSNR degraded by 0.012 dB and the bit rate increased by 0.15% on average; both results are still negligible. The average time savings increased by approximately 41.58%. We combine the two algorithms to form a two-stage algorithm for fast intra prediction coding. From the experimental results presented in Table 5, both the average degradation of the PSNR (−0.015 dB) and the average increase of the bit rate (0.30%) are minimal in our proposed two-stage algorithm, and on average, reduces the encoding time by 55.19%.

Table 4. Performance of our proposed algorithms

Test Sequence	QP	Quant-Based Block−Size Decision			Direction-Based Prediction Mode Decision			Two-Stage Approach		
		ΔPSNR (dB)	ΔBR (%)	ΔTS (%)	ΔPSNR (dB)	ΔBR (%)	ΔTS (%)	ΔPSNR (dB)	ΔBR (%)	ΔTS (%)
Container (QCIF)	16	−0.016	0.08	8.12	−0.026	0.01	38.61	−0.040	0.04	48.17
	20	−0.014	−0.30	11.19	−0.016	−0.11	38.77	−0.028	0.15	49.88
	24	−0.008	0.23	18.84	−0.026	0.16	38.25	−0.027	0.42	52.59
	28	0.002	0.26	20.38	−0.033	0.34	38.31	−0.027	0.68	53.61
	32	−0.005	0.93	23.20	−0.041	0.34	39.19	−0.048	1.80	56.59
Foreman (QCIF)	16	0.018	0.08	8.11	−0.005	0.07	41.00	0.003	0.10	50.17
	20	0.004	0.00	8.00	−0.008	0.11	40.47	−0.019	−0.09	49.71
	24	0.004	0.05	8.86	−0.007	0.10	40.83	0.005	0.32	50.48
	28	−0.005	0.08	10.87	−0.002	0.44	40.63	−0.008	0.11	51.69
	32	−0.015	0.11	14.14	−0.029	0.10	39.55	−0.034	0.50	52.48
Container (CIF)	16	0.002	0.02	19.80	−0.005	0.32	40.21	0.001	0.34	53.42
	20	0.000	0.19	22.85	−0.015	0.30	40.31	−0.010	0.34	54.79
	24	0.002	0.64	25.53	−0.017	0.42	39.62	−0.011	0.94	56.38
	28	−0.006	1.08	31.23	−0.018	0.67	39.09	−0.018	1.44	57.75
	32	−0.011	1.54	34.26	−0.016	0.45	38.66	−0.026	1.89	57.83
Hall (CIF)	16	−0.001	0.02	9.06	−0.028	−0.03	41.71	−0.030	0.05	49.50
	20	0.008	0.15	12.50	−0.004	0.03	41.58	−0.002	0.23	51.45
	24	−0.001	0.14	22.86	−0.008	0.14	41.87	−0.002	0.40	56.25
	28	−0.001	0.40	25.01	−0.008	0.17	41.39	−0.016	0.65	56.97
	32	0.006	0.41	29.12	−0.003	0.03	41.24	−0.015	0.07	58.73
Mobile (CIF)	16	−0.004	−0.03	8.13	−0.012	0.06	38.05	−0.016	0.13	45.03
	20	−0.003	−0.05	8.58	−0.012	0.10	38.10	−0.012	0.05	45.70
	24	−0.006	0.06	9.00	−0.014	0.13	38.09	−0.015	0.13	46.00
	28	−0.004	−0.01	9.85	−0.008	0.11	38.07	−0.010	0.16	46.42
	32	−0.003	−0.17	10.09	−0.011	0.00	37.48	−0.015	0.31	46.74
Paris (CIF)	16	0.001	0.10	8.29	−0.015	0.18	41.43	−0.010	0.18	48.57
	20	−0.003	−0.07	8.71	−0.007	0.12	41.35	−0.019	0.14	49.05
	24	0.006	0.06	9.24	−0.011	0.18	40.88	−0.027	0.31	49.26
	28	−0.011	−0.09	11.11	−0.018	0.19	40.97	−0.025	0.30	50.02
	32	−0.003	−0.34	14.71	−0.024	−0.05	40.40	−0.025	0.21	51.08
Parkrun (720p)	16	0.002	0.02	6.73	−0.010	0.01	39.05	−0.009	0.02	47.34
	20	0.000	0.01	15.46	−0.006	0.03	41.70	−0.008	0.01	53.38
	24	−0.001	0.04	15.16	−0.004	0.07	39.61	−0.006	0.07	51.85
	28	−0.004	−0.04	18.43	−0.006	0.06	40.45	−0.007	0.05	53.39
	32	−0.006	−0.07	23.48	−0.010	0.06	40.71	−0.009	0.11	55.11
Shields (720p)	16	−0.006	−0.06	11.36	−0.02	−0.04	41.21	−0.024	−0.11	51.19
	20	−0.003	−0.05	17.12	−0.007	0.03	41.35	−0.013	−0.04	53.50
	24	−0.001	0.12	21.82	0.003	0.42	41.81	−0.002	0.38	55.58
	28	−0.003	0.22	24.13	−0.005	0.37	41.60	−0.013	0.52	56.30

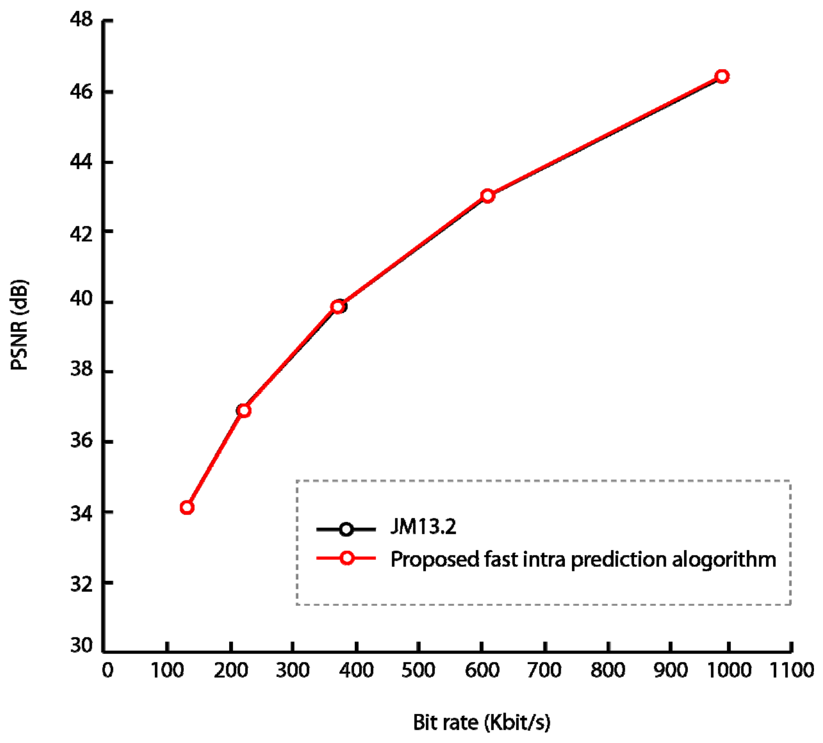
	32	−0.008	0.12	26.52	−0.016	0.33	41.21	−0.021	0.57	57.34
Blue_sky (1080p)	16	−0.016	−0.20	26.96	−0.004	0.11	38.58	−0.026	−0.16	55.87
	20	−0.002	0.02	25.62	−0.003	0.13	40.40	−0.009	0.13	56.01
	24	−0.01	0.26	32.51	−0.007	0.14	41.91	−0.017	0.39	59.09
	28	−0.006	0.32	36.18	−0.005	0.33	42.74	−0.012	0.52	60.61
	32	−0.009	0.01	35.17	−0.017	0.35	43.34	−0.020	0.30	60.82
Pedestrian (1080p)	16	−0.003	0.05	18.36	−0.027	−0.13	42.15	−0.029	−0.04	54.71
	20	−0.001	−0.04	23.79	−0.008	0.07	42.17	−0.009	0.10	56.87
	24	−0.001	0.09	30.64	−0.006	0.27	42.75	−0.007	0.38	59.66
	28	−0.002	0.14	38.60	−0.006	0.12	43.28	−0.009	0.17	62.51
	32	0.012	0.16	43.03	−0.010	−0.01	42.95	−0.007	−0.07	64.17
Station2 (1080p)	16	−0.010	−0.11	17.68	−0.019	−0.18	41.23	−0.027	−0.23	54.11
	20	−0.001	0.04	27.22	−0.003	0.17	44.29	−0.003	0.25	58.19
	24	0.000	0.04	43.54	−0.007	0.41	53.40	−0.007	0.58	66.74
	28	−0.002	0.07	47.64	−0.016	0.34	54.04	−0.015	0.33	68.48
	32	0.000	0.19	46.57	−0.020	0.24	49.71	−0.012	0.44	66.92
Sunflower (1080p)	16	−0.003	0.00	43.65	−0.011	−0.06	51.90	−0.013	0.08	67.71
	20	−0.001	−0.01	37.79	−0.011	0.17	43.70	−0.010	0.26	63.56
	24	0.000	0.06	40.42	−0.011	0.21	44.37	−0.013	0.36	64.08
	28	−0.003	−0.03	42.64	−0.023	0.14	44.08	−0.022	0.31	64.70
	32	0.007	−0.13	43.73	−0.014	0.01	43.07	−0.020	0.08	65.15
Average		−0.002	0.11	22.39	−0.012	0.15	41.58	−0.015	0.30	55.19

Table 5 shows a comparison of the average performance of our proposed algorithm with that used in reference [9]. The values in this table are the average performance values when QP = {16, 20, 24, 28}. The scope covered in the two-stage algorithm of [9] is identical to our proposed two-stage algorithm. The results show that the coding performance of our algorithm is slightly better than that of [9] in terms of the time savings (54.74% compared to 50.57%), and our proposed algorithm significantly improves the coding efficiency in terms of the PSNR degradation (−0.013 dB compared to −0.132 dB) and increased bit rate (0.25% compared to 0.90%).

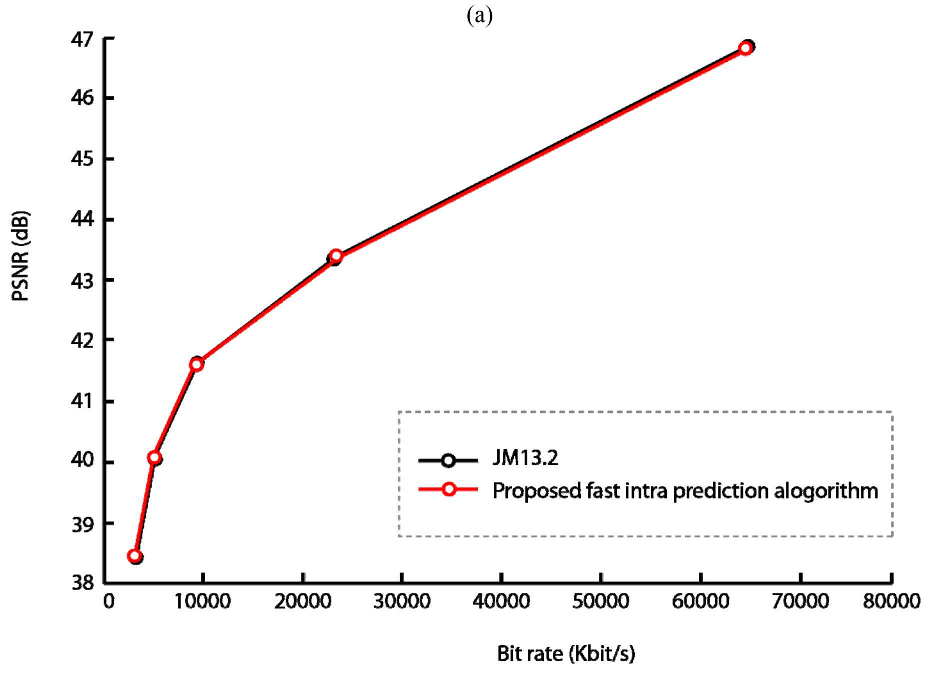
Table 5. Average performance comparison

Test Sequence	Two-Stage Approach [9]			Proposed Two-Stage Approach		
	ΔPSNR (dB)	ΔBR (%)	ΔTS (%)	ΔPSNR (dB)	ΔBR (%)	ΔTS (%)
Container (QCIF)	−0.191	0.97	48.5	−0.031	0.32	51.06
Foreman (QCIF)	−0.161	0.90	36.7	−0.005	0.11	50.51
Container (CIF)	−0.123	0.68	50.5	−0.010	0.77	55.59
Hall (CIF)	−0.145	1.12	52.6	−0.013	0.33	53.54
Mobile (CIF)	−0.154	0.57	35.5	−0.013	0.12	45.79
Parkrun (720p)	−0.147	0.61	40.8	−0.008	0.04	51.49
Shields (720p)	−0.137	0.83	44.8	−0.013	0.18	54.14
Pedestrian (1080p)	−0.089	1.10	64.7	−0.014	0.15	58.44
Station2 (1080p)	−0.097	1.04	65.6	−0.013	0.23	61.88
Sunflower (1080p)	−0.077	1.16	66.0	−0.015	0.25	65.01
Average	−0.132	0.90	50.57	**−0.013**	**0.25**	**54.74**

Figures 12(a) and (b) show the RD curves for the Mobile and Shield sequences, respectively. The RD curves achieved by our proposed algorithm nearly overlap the RD optimized curves achieved by the JM13.2. From the results, we can verify that our algorithm induces negligible RD degradation.



(a)



(b)

Fig. 12. (a) RD curves of Mobile (b) RD curves of Shields

5 Conclusion

This study presented two methods to reduce computational complexity of H.264/AVC High Profile intra prediction: quant-based block-size selection decision, and direction-based prediction mode decision. These methods form a highly efficient intra prediction algorithm. Owing to a higher HR, our proposed algorithm provides high accuracy for block size selection and mode prediction. The experimental results demonstrated that on average, the proposed algorithm reduces the encoding time by approximately 54% when compared with an exhaustive search using the JM13.2. PSNR degradation was negligible (approximately −0.013 dB on average), and the increase in bit rate was minimal (approximately 0.25% on average). As compared with the existing algorithms, the results show that our algorithm achieves a significant improvement in both computation performance and RD performance.

References

[1] ISO/IEC 14496−10, *Information Technology–Coding of Audio-Visual Objects–Part 10: Advanced Video Coding*, December 2003.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 7, pp. 560–576, 2003.

[3] I. G. Richardson, *H.264 and MPEG-4 Video Compression*, Wiley, 2003.

[4] G. J Sullivan and T. Wiegand, "Rate-distortion Optimization for Video Compression," *IEEE Signal Process. Mag.*, Vol. 15, No. 6, pp. 74–90, 1998.

[5] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G. J. Sullivan, "Rate-constrained Coder Control and Comparison of Video Coding Standards," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 13, No. 7, pp. 688–703, 2003.

[6] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, "Video Coding with H.264/AVC: Tools, Performance, and Complexity," *IEEE Circuits Syst. Mag.*, Vol. 4, No. 1, pp. 7–28, 2004.

[7] Y. W. Huang, B. Y. Hsieh, S. Y. Chien, S. Y. Ma, L. G. Chen, "Analysis and Complexity Reduction of Multiple Reference Frames Motion Estimation in H.264/AVC," *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 16, No. 4, pp. 507–522, 2006

[8] A. Yu, G. Martin, H. Park, "Fast Inter-mode Selection in The H.264/AVC Standard Using a Hierarchical Decision Process," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 18, No. 2, pp. 186–195, 2008.

[9] Y. H. Huang, T. S. Ou, H. H. Chen, "Fast Decision of Block Size, Prediction Mode, and Intra Block for H.264 Intra Prediction," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 20, No. 8, pp. 1122–1132, 2010.

[10] Y. C. Wei and C. H. Tseng, "Transformed Domain Block Size and Intra Mode Decision for Advanced Video Coding," in *Proceedings of 2010 International Symposium on Computer Communication Control and Automation (3CA)*, Vol. 1, No. 1, pp. 221–224, 2010.

[11] Y. Lin, Y. M. Lee, C. D. Wu, "Efficient Algorithm for H.264/AVC Intra Frame Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 20, No. 10, pp. 1367–1372, 2010.

[12] K. Lim, S. Kim, J. Lee, D. Pak, S. Lee, "Fast Block Size and Mode Decision Algorithm for Intra Prediction in H.264/AVC," *IEEE Trans. Consumer Electronics*, Vol. 58, No. 2, pp. 1367–1372, 2012.

[13] C. Tseng, H. M. Wang, J. F. Yang, "Enhanced Intra-4x4 Mode Decision for H.264/AVC Coders," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 16, No. 8, pp. 1027–1032, 2006.

[14] Y. M. Lee, Y. T. Sun, Y. Lin: "SATD-Based Intra Mode Decision for H.264/AVC Video Coding," *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 20, No. 3, pp. 463–469, 2010.

[15] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, S. Wu, "Fast Mode Decision Algorithm for Intra Prediction in H.264/AVC Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 15, No. 7, pp. 813–822, 2005.

[16] J. C. Wang, J. F. Wang, J. F. Yang, J. T. Chen, "A Fast Mode Decision Algorithm and Its VLSI Design for H.264/AVC Intra-prediction," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 17, No. 10, pp. 1414–1422, 2007.

[17] A. C. Tsai, A. Paul, J. C. Wang, J. F. Wang, "Intensity Gradient Technique for Efficient Intra-prediction in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 18, No. 5, pp. 694–698, 2008.

[18] A. C. Tsai, J. F. Wang, J. F. Yang, W. G. Lin, "Effective Subblock-based and Pixel-based Fast Direction Detections for H.264 Intra Prediction," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 18, No. 7, pp. 975–982, 2008.

[19] D. Y. Kim, K. H. Han, Y. L. Lee, "Adaptive Single-multiple Prediction for H.264/AVC Intra Coding," *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 20, No. 4, pp. 610–615, 2010.

[20] D. Quan, and Y. S. Ho, "Categorization for Fast Intra Prediction Mode Decision in H.264/AVC," *IEEE Trans. Consumer Electron.*, Vol. 56, No. 2, pp. 1049–1056, 2010.

[21] Y. Adibelli, M. Parlak, and I. Hamzaoglu, "Pixel Similarity Based Computation and Power Reduction Technique for H.264 Intra Prediction," *IEEE Trans. Consumer Electron.*, Vol. 56, No. 2, pp. 1079–1097, 2010.

[22] H. Zeng, K. K. Ma, and C. Cai, "Hierarchical Intra Mode Decision for H.264/AVC," *IEEE Trans. Circuit Syst. Video Technol.*, Vol. 20, No. 6, pp. 907–912, 2010.

[23] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," in *Proceedings of SPIE Conf. Applications of Digital Image Processing*, pp. 454–474, 2004.

[24] T. Wiegand, and S. Gordon, "H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas," in *Proceedings of IEEE Int. Conf. Image Processing (ICIP 2005)*, pp. 593–596, 2005.

[25] JVT reference software [Online]. Available: http://iphome.hhi.de/suehring/tml/download/

[26] G. Bjontegaard, "Calculation of Average PSNR Differences between RD Curves," in *Proceedings of 13th Meeting ITU-T Q.6/SG16 VCEG*, Austin, TX, 2001.