

An improved hybrid fast mode decision method for H.264/AVC intra coding with local information

Changnian Chen · Jiazhong Chen · Tao Xia · Zengwei Ju · Lai-Man Po

© Springer Science+Business Media New York 2013

Abstract In this paper, an improved hybrid fast mode decision method for H.264/AVC intra coding is proposed, which is based on the analysis of edge filter and more efficient mode selection. In contrast to the conventional fast mode decision methods where multiple neighbor modes are determined by the same filter, a unique filter is applied to each directional mode to eliminate the correlation among neighbor modes. To reduce the complexity, the filters are only applied to the selected pixels that cover most of the block region. After that, the results along prediction residuals of these pixels are summed for fast mode decision. Moreover, local information is also taken into account through Most Probable Mode (MPM) from neighbor-coded blocks. Experimental results demonstrated that better coding performance is achieved as compared with well-known conventional fast mode decision algorithms for H.264/AVC intra coding.

Keywords Intra prediction · Edge filter · Most probable mode · H.264/AVC

1 Introduction

Compared with previous standards, H.264/advanced video coding (AVC) achieves significant compression gain by using various advanced techniques [7, 18]. Meanwhile, these techniques lead to higher computational complexity. One of the major improvements is the use of rate-distortion (RD) optimization for intra mode decision

C. Chen · J. Chen (✉) · T. Xia · Z. Ju
School of Computer Science and Technology, Huazhong University of Science and Technology,
Wuhan 430074, China
e-mail: hust_scs@yahoo.com.cn

L.-M. Po
Department of Electronic Engineering, City University of Hong Kong,
Kowloon, Hong Kong 999077, China
e-mail: eelmpo@cityu.edu.hk

as well as inter mode decision, which check all the possible intra and inter modes to find the best coding result for achieving the highest coding efficiency. For intra prediction, it remains a challenging task to reduce its coding time while retaining the coding performance. The intra coding supports three macroblock partitions: 4×4 , 8×8 , 16×16 , with a certain number of prediction modes. Selecting fewer block-size types and prediction modes is normally a good solution to the task. As larger block sizes are fit for smooth regions whereas smaller block sizes are suitable for rich contents, block-size types are always reduced by thresholds yielded according to the coding content. However, things become complicated when efforts are made to reduce the prediction modes in each block-size type for the best mode that is not only with smallest distortion but also fewer coding bits. This is not straightforward to predict the required bitrate precisely, as the bitrate are obtained by complicated procedures: transformation, quantization, inverse transformation, inverse quantization, and entropy coding. In addition, the entropy coding of H.264/AVC involves context information which making the prediction more sophisticated [10]. Rate estimation is still under research and cannot be directly used for fast intra mode decision [5]. In this paper, we focus on edge detection based distortion prediction for realizing fast intra mode decision of H.264/AVC.

The use of edge detection in intra coding has been well elaborated in a number of literatures [2, 3, 6, 9, 13, 14, 16, 17, 19]. Most of the edge detection methods utilize the local property of video content to investigate the best prediction direction for intra coding. Variety of edge filters are designed in those methods. The accuracy of mode direction prediction depends on the edge filters. Thus it is important to set each direction with unique edge filter, which have been done in our previous work [2]. More details about the edge filters of those methods will be given in the Section 2.

In this work, we present an improved hybrid mode decision method to further reduce its computational requirement while maintaining its rate-distortion performance. In the proposed method, the edge filtered results are summed with the prediction residuals according to the modes they represent, which can select the probable modes efficiently and avoid the sort operations at the stage of individual pixel processing. Inspired by [6], we also incorporate MPM (Most Probable Mode) that is proved to be useful in the proposed method. Our experimental results show that the proposed algorithm outperforms its previous rivals in terms of coding efficiency.

The rest of this paper is organized as follows. In Section 2, we give a brief review of H.264/AVC intra coding and the related edge detection methods in the literatures. The proposed algorithm is introduced in Section 3. Experimental results and discussion are presented in Section 4, followed by a brief conclusion in Section 5.

2 Review of the H.264/AVC intra coding and related works

2.1 Intra coding scheme in H.264 standard

In this subsection, the intra coding of H.264/AVC [18] is briefly reviewed for supporting the presentation of the proposed method. For the luma samples, H.264/AVC supports three block-size types in high profile: luma 4×4 (I4MB), luma 8×8 (I8MB), and luma 16×16 (I16MB). I4MB supports eight directional modes as well as DC mode as shown in Fig. 1. I8MB shares the same prediction modes with

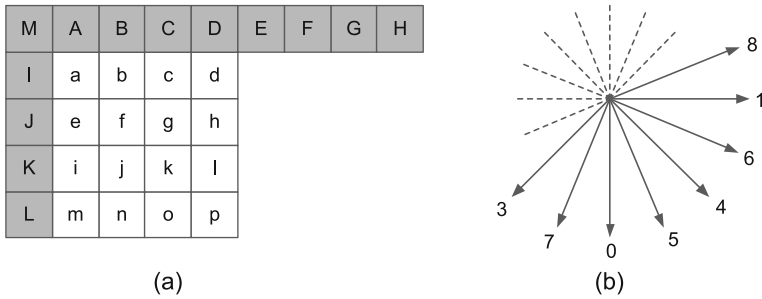


Fig. 1 **a** A 4×4 block and its referenced neighbor pixels. **b** Eight directional modes for I4MB and I8MB prediction

I4MB except that the coding block size is 8×8 . I16MB contains three modes with directions: vertical, horizontal, and plane. In this work, plane mode is simply treated as the diagonal down-left mode that is the mode 3 as shown in Fig. 1b. For the chroma intra prediction, only 8×8 block (C8MB) is supported and the prediction modes are the same as those of I16MB.

2.2 Related works on the design of edge filters

In [13], Sobel operator that comprises two 3×3 filters as shown in Fig. 2a is convolved with all the pixels to create an edge direction histogram. For Intra 4×4 , the mode with the maximum amplitude in the histogram together with its neighbor modes and DC mode are selected for intra coding. In [3], Sobel operator is replaced by Roberts cross as shown in Fig. 2b and the number of pixels processed is reduced by the cross. These two algorithms introduce the widely used edge detection approach to H.264/AVC coding. However, they do not involve the intra prediction scheme of H.264/AVC. A gravity approach is introduced in [4], in which the direction of the block’s gravity center is used to indicate the candidate modes. The pixels of a block are considered as a set of material points and the intensities of these pixels are regarded as pixel mass to generate a unique location that denoted as the gravity center of the block. Figure 3 shows the edge filters with size of 2×2 used in [17]. Unlike filters used for seeking the minimal value, the mode of maximal value is chosen in a set of filters that are perpendicular to the prediction directions. When the

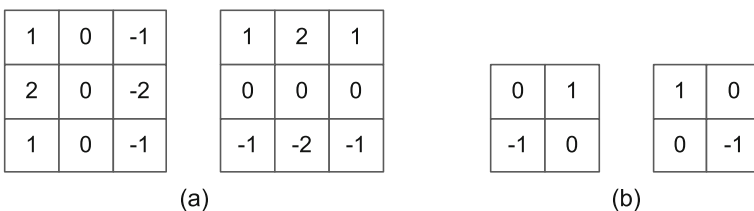


Fig. 2 Edge operators used in [3, 13]. **a** Sobel and **b** Roberts

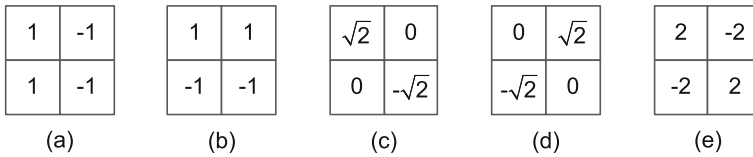


Fig. 3 Five filters of edge operator used in [17]. **a** Vertical. **b** Horizontal. **c** Diagonal down-left. **d** Diagonal down-right. **e** Nondirection

maximum corresponds to the 5th filter operation, all the possible modes should be calculated. These filters are applied only once to a pseudo 2×2 block. Tsai et al. [16] present an intensity gradient approach for intra prediction with use of the filters as shown in Fig. 4. Four filters are designed for the first four prediction directions along the prediction orientations while the other four are generated through being set to be the average of their two neighbor modes. The filters are calculated with the central four pixels in a 4×4 block, such as pixels f, g, j, and k of the block in Fig. 1a. However, it is insufficient to fully extract the entire block information with only four pixels. The algorithm introduced in [14] handles the filters with all the 16 pixels while the filters shown in Fig. 5 are slightly changed as well. When processing all the pixels to a 4×4 block, the most probable directions are determined through a set of sort operations. In our previous work [2], we build the correspondence between modes and filters which is shown in Fig. 6. Unlike the filters in [14, 16], a unique filter is applied to each directional mode to eliminate the correlation among neighbor modes. However, we adopt a computational expensive scheme, which is the same as that in [14]. The recently proposed fast decision methods have demonstrated how to reduce the block-size types as well as prediction modes. For instance, in [19], a hierarchical intra mode decision method is proposed based on the sum of absolute Hadamard transform difference (SAHTD) and quantization parameter (QP) dependent threshold. The algorithm in [6] utilizes the variance-based classification of texture complexity to make the block-size type selection and combines the fast mode algorithm in [17] with MPM to make the fast mode decision. Lee and Jang [9] propose another block-size type selection method in which not only the variances of input pixels but those of prediction residual are taken into account. Prediction residual derived from intra 16×16 coding is supposed to be related to the RD performance. Thresholds are carefully determined for precise selection. For the mode selection part, the principle used is normally similar to the existing fast mode decision methods.

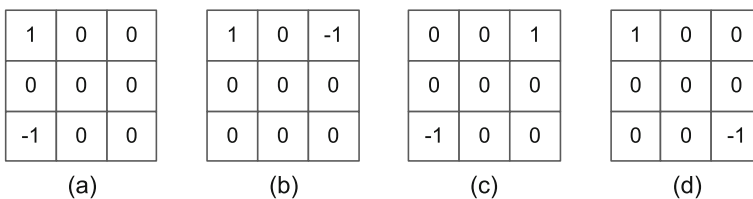


Fig. 4 Filters used in [16]. **a–d** are the filters for the first four directional modes

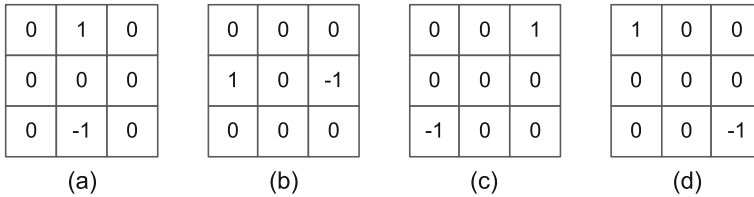


Fig. 5 A slightly modified version of filters in [16] used in [14]

2.3 The hybrid method

Recently, edge filters are adopted in the hybrid method and the filters used in [2] are shown in Fig. 6. These filters are performed in a different manner as compared with Sobel operator used in [13]. Take pixel f of Fig. 1a as an example. When Sobel operator is applied, the degree of difference in vertical direction D_x is represented as

$$D_x = |a - i + 2 \times (b - j) + c - k| \tag{1}$$

For Sobel operator, the subtraction operations are first performed along the direction and then the sum is obtained. The absolute value of the sum represents the directional mode. To better represent the dominant edge, a calculation order is adopted in [16]. That is, the absolute value of the subtraction along the direction is obtained first, and the mean of all the absolute values is the value for comparison. For instance, when the filters as shown in Fig. 6 are applied, the vertical mode indicated by D_0 is defined by

$$D_0 = \frac{1}{2}(|a - i| + |c - k|) \tag{2}$$

The pixels involved in this process are shown in Fig. 7. These pixels are processed with the edge filters to conduct the values corresponding to directional modes. These

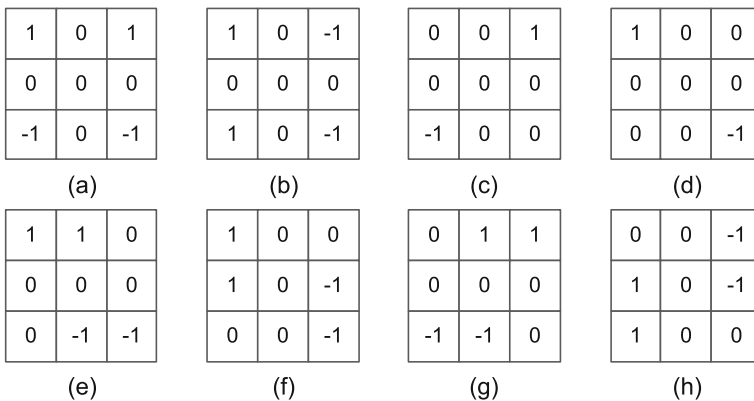


Fig. 6 Filters used in the hybrid method [2]. **a-h** correspond to the eight directional modes (m0–m8) supported in luma 4×4 and 8×8 block

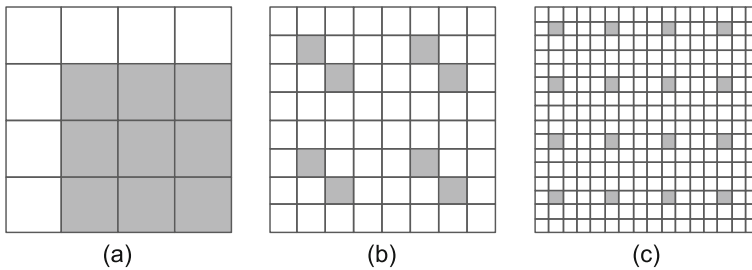


Fig. 7 Pixels selected in the hybrid method [2]. **a** 4×4 , **b** 8×8 , and **c** 16×16

values are summed with the prediction residuals. Each pixel in the block has the same number of directional prediction values generated by prediction process and the prediction residuals can be obtained by subtracting them with their original values.

The prediction modes selected for intra coding are listed in Table 1. Take I4MB as an example, the mode decision depends on a statistical scheme. Each pixel will conduct a serial of values corresponding to directional modes. Small value means there is less variance along that direction and it should be selected. Sort operation is adopted to obtain two minimal values and the modes they represent are counted. After processing all the pixels, the three modes with the highest frequency of appearance are chosen for intra coding along with the DC mode. Other block-size types are performed in the similar way. In the hybrid method, 3 out of 8 directional modes are selected for I4MB and I8MB while only one directional mode is selected for those block-size types with 3 directional modes. As for chroma prediction, the same prediction mode is applied to the two 8×8 blocks (U and V). If the result of each block calculated is different, only DC mode is set to be the selected mode, which can further reduce the coding time. For blocks at the frame boundary where the left or upper block may not be available, only those prediction modes available are taken into consideration.

3 The proposed method

To derive useful information from local blocks, it is preferable to choose filters with smaller sizes. For I4MB and I8MB with 8 directional modes, the filter size is at least 3×3 if a unique filter is set for each mode. In this work, the edge filters shown in Fig. 6 are chosen through comprehensive experiments on filter selection for each block-size type. The first three filters are used to represent directional modes (i.e. horizontal, vertical, and plane) for I16MB and C8MB. Plane mode is considered as diagonal down-left mode.

Table 1 The number of selected modes of each algorithm

	I4MB	I8MB	I16MB	C8MB
Tsai et al. [16]	4	4	2	2
Gravity	2	1–5	2–4	1–2
Hybrid	4	4	2	1–2
Proposed	2–5	2–5	2	1–2

Hybrid method has shown to achieve better performance than conventional edge detection methods by adapting these filters. In this work, not only this advantage is utilized, several other improvements are also investigated. First, the sort operations at the stage of individual pixel processing are time consuming and summing method is used instead. Second, as we adapt computation efficient summing method, more pixels can be applied to improve the filtering results with negligible time increase. Finally, local information is exploited to enhance the mode selection accuracy. These improvements are illustrated in details in the following subsections.

3.1 Applying the filters to coding blocks

After generating the filters for different macroblock partitions, it comes to the problem of how to apply them to pixel level. Intuitively, it should filter all the pixels within the coding block. However, this will significantly increase the computational requirement. Our empirical experimental results show that it is a better choice to apply the filters to several distributed pixels that cover most of the block region, i.e., the gray regions in Fig. 8. Each pixel will conduct with one filter data for each directional mode. When all selected pixels are computed, the statistical data of each directional mode are obtained. Then it is crucial to process these data for accurate decision. In [2], two directional modes with the least values are chosen from each pixel. After that, the three directional modes that appear most frequently in all the calculated pixels are finally selected for intra coding. In I4MB, when edge filters are applied to one pixel, 8 filtering values can be generated and each value corresponds to one prediction direction. Small value means there is less variance along that direction and it should be selected. Thus, these 8 values are sorted and the first two minimal ones are chosen, which are the direction modes they represent will be counted. However, the sort operation tends to ignore the variance of those uncounted values. When there are noise-like pixels, sort operation takes them as normal pixels and as a result, their influence on mode decision cannot be revealed. Moreover, their filtering results will play more important roles than those of other normal pixels through summing process. It is noticed that when summing all the data together according to the modes they represent, better performance can be achieved and the sort operations can be avoided during individual pixel processing.

As for the selected pixels shown in Fig. 8, although up to 64 pixels are chosen for I16MB, the additional computational load is negligible with the summing method.

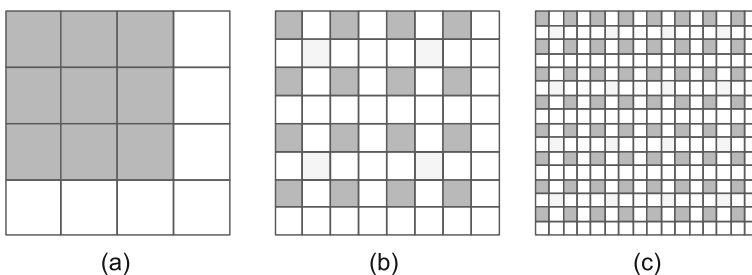


Fig. 8 Pixels used for calculation with filters. **a** 4×4 , **b** 8×8 , and **c** 16×16

In summary, the selection of directional modes in this work can be described as following:

First, the edge filters are applied to the selected pixels for $n(n = 1, 2, \dots, N)$ in order to obtain the filtered result of D_m^n . The prediction residual P_m^n with $m = 1, 2, \dots, M$ can also be acquired, where N represents the number of selected pixels and M is the number of directional modes used in this block.

After that, values of all the selected pixels are accumulated and the mean of them is summed as:

$$H_m = \frac{1}{N} \sum_{n=1}^N D_m^n + \frac{1}{N} \sum_{n=1}^N P_m^n$$

(3)

for $m = 1, 2, \dots, M$ and $n = 1, 2, \dots, N$

It is more probable for the directional modes with lower H to be selected for intra coding. Prediction residuals are also involved in this work as they can reduce the degradation in coding performance, which is partly due to the intra prediction scheme in H.264/AVC. Although prediction modes with directional features serve as means for edge detection, the prediction processes are not identical for all the directions. For example, both the upper and the left reconstructed reference pixels can be used in mode 4 whereas mode 3 only uses the upper reference pixels as illustrated in the H.264/AVC standard. When the upright block is unavailable, more than half the pixels in the coding block are predicted only by the last one of the upper reference pixels. As a result, it is insufficient for the original pixels alone to represent the prediction mode in that direction for comparison. The directional mode with the minimal value should have smaller filtering result along the direction as well as prediction residual.

3.2 Combined with local information

In the conventional edge detection based fast mode decision algorithms, DC mode is taken as the default mode and it is always computed in intra coding. Recently, MPM is introduced to utilize the local relationship between neighbor blocks. Moreover, the belief propagation technique to infer the potentially best mode of current block is adopted in [12], in which the best mode of coded block is refined by passing the information forward and backward among the local blocks. In our proposed method, MPM is fully exploited to instruct the mode selection procedure. For the blocks with all prediction modes available in I4MB and I8MB, the MPM can be any of the nine prediction modes, which is generated through the comparison of the best prediction modes of the upper and left blocks. Then fast mode decision is made according to MPM. If MPM has high probability to be the best mode, fewer modes are employed for intra coding; otherwise more candidate modes should be considered for prediction accuracy. In other words, MPM will be firstly selected while other modes will be chosen or not concerning their importance to MPM. The relationship between MPM and the best mode in the case of I4MB is given in Table 2. Similar properties can be found in I8MB and we take I4MB as an example for convenience. In the Table 2, $H_i(i = 0 \dots 7)$ depicts the sorted directional modes obtained by (3). BM represents the best mode determined by H.264/AVC. From this table, we can find that H_0 is the best choice for nearly half of those blocks. The larger the filtering result

Table 2 The number (Num) and percentage (%) of each mode to be the best mode (BM), MPM, and both of most probable mode and best mode (MPBM) in case of 14MB and QP = 28 for encoding 300 frames

	Foreman						Mobile					
	BM		MPM		MPBM		BM		MPM		MPBM	
	Num	%	Num	%	Num	%	Num	%	Num	%	Num	%
H_0	212,135	46.98	116,370	25.77	95,279	43.19	864,675	46.66	496,157	26.77	381,084	51.95
H_1	72,959	16.15	58,966	13.06	28,464	12.90	300,838	16.24	228,704	12.34	92,425	12.60
H_2	36,778	8.15	43,321	9.59	16,374	7.42	161,768	8.73	166,583	8.99	50,229	6.85
H_3	23,061	5.11	35,651	7.89	10,592	4.80	111,949	6.04	157,052	8.47	38,474	5.24
H_4	16,482	3.65	32,417	7.18	8,207	3.72	82,352	4.44	146,232	7.89	29,854	4.07
H_5	11,046	2.45	34,146	7.56	5,768	2.61	59,767	3.23	148,376	8.01	20,592	2.81
H_6	7,585	1.68	21,849	4.83	3,868	1.75	43,226	2.33	121,213	6.54	13,041	1.77
H_7	5,421	1.20	18,231	4.04	2,523	1.14	33,359	1.80	114,901	6.20	8,689	1.18
DC	66,033	14.63	90,549	20.08	49,536	22.47	195,166	10.53	273,882	14.79	99,191	13.53
Total	451,500	100	451,500	100	220,611	100	1,853,100	100	1,853,100	100	733,579	100

H is, the less possibility the mode has to be the best one. In [6], three directional modes are always selected and DC mode is chosen only when MPM is one of the three directional modes. It is not an appropriate scheme for the edge filtering results as it is shown in Table 2 that DC mode is more frequently chosen to be the best mode than H_2 . Moreover, to further reduce the coding time while not degrading the coding performance, an adaptive number of selected modes are more suitable than the fixed four candidate modes. For example, if MPM is H_0 , then it is highly possible to be the best mode and fewer modes will be selected. The situation when MPM fails to be the best mode is studied, which is shown in Table 3. It indicates other candidate modes that should be considered and the column shows the amount of other modes as the best mode. The number of selected modes for each condition is described as follows.

- (1) MPM is H_0 . Over one third of those blocks with different best modes are H_1 . As a consequence, to further reduce the coding time, no more other modes will be selected in this case to save much time with little performance degradation. From Table 3, we can see that H_0 is highly recommended for selection. Therefore, when MPM is other modes, H_0 is still selected.
- (2) The second largest amount of MPM is DC. Unlike (1), nearly half of MPM are not the best choices in sequence “Foreman” and the amount becomes even smaller when video material is with plenty of details such as “Mobile”. Thus more directional modes should be selected and besides H_0 , H_1 is included. Considering the large amount of MPM in this case, H_2 is not selected for computation time’s sake.
- (3) MPM is H_1 . In this case, most of the best modes are H_0 if are not MPM. We make the similar choice to (2), selecting H_0 , H_1 , and DC for intra coding.
- (4) MPM is H_2 . Directional modes ahead of it will be chosen besides DC, which is the same as [2, 6].
- (5) MPM is of other directional modes. In this case, three modes, i.e. H_0 , H_1 , and H_2 are taken into account together with DC and MPM. Although an additional one is selected compared with the method in [6], more coding gain is achieved with relatively little coding time increase.

The 8 directional modes for I4MB and I8MB mode selections are listed in Table 4. They are changeable for different requirements. For faster intra coding, the selected modes of those blocks with MPM equal to $H_i (i = 3 \dots 7)$ can be further reduced; for

Table 3 Modes that is the best mode but not MPM in the case of I4MB for the “Foreman” sequence in QCIF format

	H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7	DC
H_0	0	21,597	15,388	13,426	12,348	14,822	8,120	6,593	24,562
H_1	8,878	0	5,846	5,097	4,761	5,374	3,656	3,124	7,759
H_2	3,349	2,771	0	2,369	2,282	2,672	1,888	1,680	3,393
H_3	2,215	1,396	1,379	0	1,587	1,673	1,119	1,061	2,039
H_4	1,453	980	881	897	0	1,092	790	778	1,404
H_5	925	662	620	597	573	0	563	531	807
H_6	618	475	423	380	369	435	0	479	538
H_7	507	344	296	315	348	292	285	0	511
DC	3,146	2,277	2,114	1,978	1,942	2018	1,560	1,462	0
Total	21,091	30,502	26,947	25,059	24,210	28,378	17,981	15,708	41,013

Table 4 Other selected modes based on MPM for I4MB and I8MB

MPM	Other selected modes	Total number
H_0	H_1	2
H_1	H_0, DC	3
DC	H_0, H_1	3
H_2	H_0, H_1, DC	4
H_3-H_7	H_0, H_1, H_2, DC	5

MPM is always selected

less coding performance loss, H_2 can be included among those blocks when MPM is equaled to H_1 or DC. Mode selection for I16MB and C8MB is the same as hybrid method. For I16MB, only one directional mode will be selected according to the filtering results in addition to DC mode. In the case of C8MB, if U and V conduct the same directional mode, then this mode along with DC mode will be selected. Otherwise, only DC mode will be computed.

4 Experimental results and discussion

The proposed algorithm was implemented on JM 18.3 provided by JVT [8]. The experimental settings are based on recommended simulation common conditions provided in [15]. More details of the configurations are as follows.

- (1) All the frames are encoded as intra frames.
- (2) RD optimization and CABAC are enabled.
- (3) QPs are set to be 22, 27, 32, and 37.
- (4) RD optimization quantization and fast chroma prediction are turned off.

The performance evaluation is made with respect to the JM reference software based on the following performance measures: the peak signal-to-noise ratio (PSNR) changes $\Delta PSNR$ (in decibel), the total bitrate (BR) increases ΔBR (in percentage), and the time saving ΔT (in percentage). The average PSNR value of luma (Y) and chroma (U, V) is used, which is based on the equations below [13]:

$$\overline{PSNR} = 10 \log_{10} \left(\frac{255^2}{\overline{MSE}} \right) \tag{4}$$

where the average mean square error (MSE) is given by

$$\overline{MSE} = \frac{4 \times MSE_Y + MSE_U + MSE_V}{6} \tag{5}$$

Table 5 The comparison of sort based approach and summing method

Sequence	Sort	$\Delta PSNR$ (dB)	ΔBR (%)	ΔT (%)
		QCIF	Foreman	-0.002
	News	-0.008	0.15	1.10
CIF	Paris	-0.004	0.20	1.21
	Mobile	-0.007	0.12	1.59
	Average	-0.005	0.19	1.32

The results here are acquired with respect to the proposed method shown in Table 6

Table 6 Performance comparison

Sequence	Tsay et al. [16]			Wang et al. [17]			Huang et al. [6]			Hybrid			Gravity			Lee et al. [9]			Proposed			
	Δ PSNR (dB)	Δ BR (%)	Δ T (%)	Δ PSNR (dB)	Δ BR (%)	Δ T (%)	Δ PSNR (dB)	Δ BR (%)	Δ T (%)	Δ PSNR (dB)	Δ BR (%)	Δ T (%)	Δ PSNR (dB)	Δ BR (%)	Δ T (%)	Δ PSNR (dB)	Δ BR (%)	Δ T (%)	Δ PSNR (dB)	Δ BR (%)	Δ T (%)	
QCIF																						
Foreman	-0.055	4.61	-63.0	-0.178	1.14	-28.6	-0.143	0.79	-40.2	-0.030	1.61	-72.7	-0.153	1.89	-74.6	-0.054	0.93	-75.4	-0.037	0.55	-74.6	
News	-0.073	4.34	-64.4	-0.183	1.09	-30.0	-0.144	0.89	-41.7	-0.038	1.42	-73.3	-0.155	1.67	-76.3	-0.072	0.71	-76.3	-0.056	0.30	-75.6	
Silent	-0.069	5.06	-63.8	-0.192	0.95	-27.1	-0.158	0.86	-39.5	-0.035	1.34	-73.0	-0.134	1.30	-77.7	-0.069	0.89	-76.1	-0.053	0.41	-74.4	
CIF																						
Coastguard	-0.120	4.33	-64.9	-0.179	0.92	-29.4	-0.094	0.79	-50.8	-0.045	1.08	-73.1	-0.149	2.25	-81.2	-0.095	0.72	-76.9	-0.055	-0.04	-75.6	
Container	-0.093	3.67	-63.7	-0.165	1.29	-30.2	-0.110	0.71	-52.5	-0.049	1.69	-72.7	-0.155	1.62	-80.1	-0.082	0.86	-78.9	-0.050	0.41	-76.5	
Mobile	-0.117	1.60	-64.8	-0.154	0.76	-28.6	-0.105	0.59	-46.3	-0.089	0.72	-74.2	-0.178	1.39	-83.3	-0.147	0.52	-78.1	-0.087	0.13	-76.6	
Paris	-0.062	5.26	-64.5	-0.170	1.02	-27.6	-0.089	0.81	-50.7	-0.042	1.15	-74.6	-0.093	2.37	-79.0	-0.059	0.94	-77.4	-0.056	0.37	-77.1	
720p																						
Parkrun	-0.162	7.96	-66.2	-0.134	0.85	-31.7	-0.114	0.76	-52.0	-0.077	0.65	-75.2	-0.165	1.21	-83.5	-0.114	0.59	-82.3	-0.087	0.04	-80.3	
Stockholm	-0.105	8.61	-65.7	-0.112	0.78	-32.6	-0.094	0.63	-55.7	-0.046	1.13	-74.7	-0.134	2.53	-82.1	-0.090	1.88	-83.9	-0.050	0.40	-79.1	
1080p																						
Pedestrian	-0.123	13.72	-67.9	-0.118	1.24	-35.3	-0.091	1.02	-65.6	-0.049	2.76	-76.8	-0.102	3.65	-84.5	-0.031	3.11	-84.5	-0.041	0.81	-81.1	
Station2	-0.137	9.16	-68.4	-0.107	1.06	-34.8	-0.102	0.87	-66.2	-0.035	2.31	-77.1	-0.129	3.57	-85.4	-0.075	2.34	-83.5	-0.046	0.35	-81.6	
Sunflower	-0.092	11.53	-68.8	-0.098	1.25	-37.1	-0.075	1.68	-67.5	-0.052	3.96	-76.1	-0.155	4.96	-84.8	-0.086	3.52	-85.1	-0.059	1.45	-80.3	
Average	-0.101	6.65	-65.5	-0.149	1.03	-31.1	-0.110	0.87	-52.4	-0.049	1.65	-74.5	-0.142	2.37	-81.1	-0.081	1.42	-79.9	-0.056	0.43	-77.7	

Table 7 The performance of proposed method with QP = 22, 27, 32, 37

Sequence	QP = 22			QP = 27			QP = 32			QP = 37		
	Δ PSNR (dB)	Δ BR (%)	TR (second)	Δ PSNR (dB)	Δ BR (%)	TR (second)	Δ PSNR (dB)	Δ BR (%)	TR (second)	Δ PSNR (dB)	Δ BR (%)	TR (second)
CIF	-0.070	-0.02	170,380	-0.050	-0.01	144,611	-0.057	-0.02	124,331	-0.044	-0.10	107,458
Mobile	-0.107	0.17	206,242	-0.088	0.23	177,558	-0.083	0.02	151,866	-0.071	0.10	127,744
Station2	-0.044	0.59	2,753,811	-0.046	0.49	2,449,472	-0.047	0.44	2,167,706	-0.055	0.13	2,023,853
Sunflower	-0.040	1.57	2,529,785	-0.054	1.35	2,379,175	-0.066	1.47	2,178,403	-0.075	1.43	2,066,368
Average	-0.065	0.58	1,415,055	-0.060	0.52	1,287,704	-0.063	0.48	1,155,577	-0.061	0.39	1,081,356

The summing method depicted in (3) is first compared with the sort based method in [2] using same edge filters and other test conditions. The comparison results are listed in Table 5, which show that the summing method can achieve better RD performance with lower computational requirement as compared with the sort

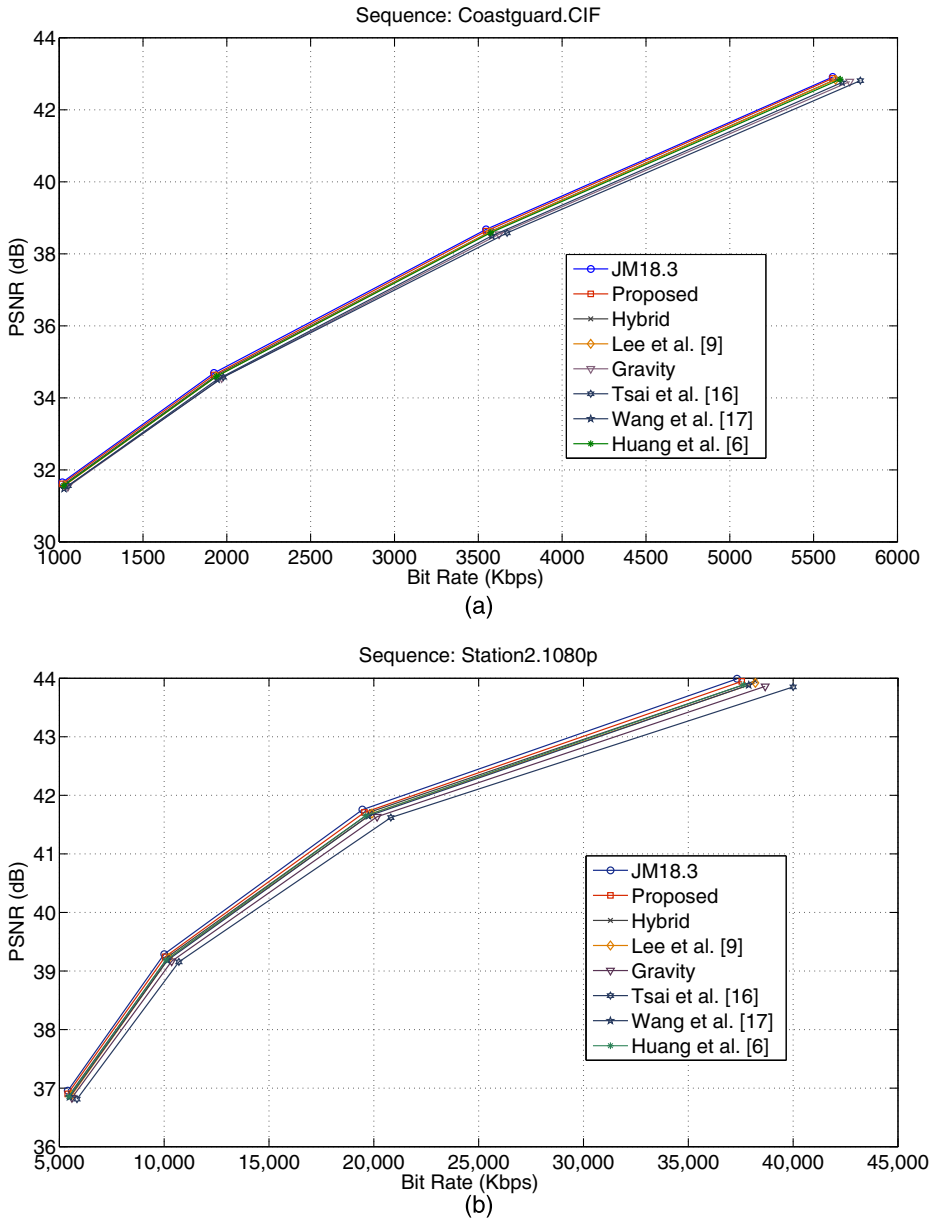


Fig. 9 RD curves compared to existing algorithms for sequences: **a** Coastguard in CIF, and **b** Station2 in1080p

based method. Consequently, we incorporate the summing method into the proposed method for obtaining better coding performance.

Table 6 compares the proposed method with Tsai et al.'s method [16], Wang et al.'s method [17], Huang et al.'s method [6], gravity method [4], Lee et al.'s method [9], and hybrid method [2]. The experimental results show the proposed algorithm can achieve around 0.09dB PSNR gain, 2 % bit rate reduction, and 4 % encoding time increase compared with [4]. Based on the statistic of Table 1, it can be seen that the time reduced by gravity method is primarily due to the fact that only one directional mode is selected in I4MB. Although it further reduces the encoding time, it inevitably degrades the coding gain as well. The proposed method also can achieve 0.03dB PSNR gain, 1 % bit rate reduction, and similar encoding time reduction compared with [9]. The proposed method outperforms Tsai et al.'s method [16] in all respects. In comparison with Wang et al. [17] and Huang et al. [6], the proposed method achieves more time reduction, less PSNR drop and bitrate increase. Also, the proposed method outperforms the previous work [2], which further demonstrates the efficiency of using summing and MPM techniques in the proposed method. In general, the proposed method is more preferable for intra mode decision. In addition, the performances of the proposed algorithm under different QPs are listed in Table 7. *TR* measured in second represents time reduction that the proposed method achieved. The RD curves for “Coastguard” and “Station2” sequences are depicted in Fig. 9 [1]. The figure demonstrates that the curves of [9] and hybrid method are very close to each other while the proposed method is closer to the JM standard than all the other methods, especially in the case of coding 1080p sequence, which illustrates that the proposed method is suitable for High Definition (HD) sequences.

It is known that the intra-mode in H.264/AVC, which utilizes spatial prediction and is transformed by discrete cosine transform (DCT), is comparable to wavelet-based still image coding standard Joint Photographic Experts Group (JPEG) 2000 [11]. Hence, the proposed improved block-based hybrid coding scheme is of significant benefit to still-image coding as well.

5 Conclusion

In this paper, an improved hybrid fast intra mode decision method for H.264/AVC is presented, in which the computational consuming sort procedure is replaced by summing method and local information is incorporated into MPM. Extensive experimental results show that the proposed method can achieve 77 % total encoding time reduction with only 0.05 dB PSNR degradation and 0.4 % bit rate increase on average. This performance is more efficient than most of the well-known fast intra mode decision algorithms for H.264/AVC.

References

1. Bjontegaard G (2001) Calculation of average PSNR differences between RD-curves. ITU-T Q.6/SG16 VCEG, document VCEG-M33.doc
2. Chen C, Chen J, Ouyang K, Xia T, Zhou J (2013) A hybrid fast mode decision method for H.264/AVC intra prediction. *Multimed Tools Appl* 62(3):719–731

3. de-Frutos-López M, Orellana-Quirós D, Pujol-Alcolado JC, Díaz-de-María F (2010) An improved fast mode decision algorithm for intraprediction in H.264/AVC video coding. *Signal Process Image Commun* 25(10):709–716
4. Elyousfi A (2013) Gravity direction-based ultra-fast intraprediction algorithm for H.264/AVC video coding. *Signal Image Video Process* 7(1):53–65
5. Hahm J, Kyung CM (2010) Efficient CABAC Rate Estimation for H.264/AVC Mode Decision. *IEEE Trans Circuits Syst Video Technol* 20(2):310–316
6. Huang YH, Ou TS, Chen HH (2010) Fast decision of block size, prediction mode, and intra block for H.264 intra prediction. *IEEE Trans Circuits Syst Video Technol* 20(8):1122–1132
7. ITU-T, JVT (2003) Draft ITU-T recommendation and final draft international standard of joint video specification (ITUT Rec.H.264/AVC/ISO/IEC 14496-10 AVC). Doc.JVTG050. Joint Video Team (JVT) of ISO/IEC MPEAG and ITUT VCEG
8. Joint Video Team Reference Software JM18.3. Available <http://iphome.hhi.de/suehring/tml/download/>
9. Lee S, Jang ES (2011) Fast intramode decision using residual variance in AVC/H.264. *Opt Eng* 50(11):117402
10. Marpe D, Schwarz H, Wiegand T (2003) Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Trans Circuits Syst Video Technol* 13(7):620–636
11. Marpe D, George V, Cycon HL, Barthel KU (2004) Performance evaluation of motion-JPEG2000 in comparison with H.264/AVC operated in pure intracoding mode. In: Frederic T (ed). SPIE, pp 129–137
12. Milani S (2011) Fast H.264/AVC FRExt intra coding using belief propagation. *IEEE Trans Image Process* 20(1):121–131
13. Pan F, Lin X, Rahardja S, Lim KP, Li ZG, Wu DJ, Wu S (2005) Fast mode decision algorithm for intraprediction in H.264/AVC video coding. *IEEE Trans Circuits Syst Video Technol* 15(7):813–822
14. Su X, Ji L, Li X (2011) A fast and low complexity approach for H.264/AVC intra mode decision. *Multimed Tools Appl* 52(1):65–76
15. Tan T, Sullivan G, Wedi T (2008) Recommended simulation common conditions for coding efficiency experiments revision 2. ITU-T VCEG, Input/Dis. VCEG-AH10r3, Antalya, Turquie
16. Tsai AC, Paul A, Wang JC, Wang JF (2008) Intensity gradient technique for efficient intraprediction in H.264/AVC. *IEEE Trans Circuits Syst Video Technol* 18(5):694–698
17. Wang JC, Wang JF, Yang JF, Chen JT (2007) A fast mode decision algorithm and its VLSI design for H.264/AVC intra-prediction. *IEEE Trans Circuits Syst Video Technol* 17(10):1414–1422
18. Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. *IEEE Trans Circuits Syst Video Technol* 13(7):560–576
19. Zeng HQ, Ma KK, Cai CH (2010) Hierarchical intra mode decision for H.264/AVC. *IEEE Trans Circuits Syst Video Technol* 20(6):907–912



Changnian Chen is now pursuing the Ph.D. degree in computer science at Huazhong University of Science and Technology (HUST). His main interests are image processing and video coding.



Jiazhong Chen received his M.S. degree in computation mathematics in 1999 and Ph.D. degree in computer system architecture in 2003 from HUST. He is now an Associate Professor of School of Computer Science and Technology of HUST. His main research interests include signal processing and wavelet analysis, image and video coding, distributed and parallel computing.



Tao Xia received his Ph.D. degree in computer system architecture in 2001 from HUST. He is now a lecturer of School of Computer Science and Technology of HUST. His main research interests include mobile computing, image and video coding.



Zengwei Ju is currently working toward a PhD at HUST. His main interests are video communication and opportunistic network.



Lai-Man Po (M'92) received the B.Sc. and Ph.D. degrees, both in electronic engineering, from the City University of Hong Kong in 1988 and 1991, respectively. Since November 1991, he has been with the Department of Electronic Engineering, City University of Hong Kong, and is currently Associate Professor. His research interests are in vector quantization, motion estimation for video compression and H.264/AVC fast mode decision algorithms.