www.ietdl.org

Published in IET Communications
Received on 6th September 2013
Revised on 27th January 2014
Accepted on 6th February 2014
doi: 10.1049/iet-com.2013.0797

ISSN 1751-8628

Complexity scalable intra-prediction mode decision algorithm for mobile video applications

Yun Song[1,2], Jizhen Long[2], Kun Yang[3], Gaobo Yang[1]

[1]School of Information Science and Engineering, Hunan University, Changsha 410082, People's Republic of China
[2]School of Computer and Communication Engineering, Changsha University of Science & Technology, Changsha 410004, People's Republic of China
[3]School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK
E-mail: yanggaobo@hnu.edu.cn

Abstract: The full search scheme employed in H.264/AVC significantly improves the coding performance, but it also introduces a very high computational complexity which limits the applications in resource-constrained mobile devices. In this study, the authors firstly present a discretisation total variation and orientation gradient-based hierarchical intra-prediction mode decision method for mobile video applications. By shrinking the candidate mode set in the rate–distortion optimisation (RDO) process, the proposed algorithm reduces the computational complexity and power consumption of the encoder. Furthermore, they extend the hierarchical algorithm to a complexity scalable version in which the coding complexity is measured on five levels by reserving various numbers of modes for RDO. Experimental results demonstrate that the proposed mode decision algorithm reduces the coding complexity significantly with negligible performance degradation and the proposed complexity scalable algorithm is effective and efficient for mobile video application.

1 Introduction

The international video coding standard, H.264/AVC or MPEG4 part 10 [1] developed by the joint video team (JVT) and formed by ITU-TVCED and ISO/IEC MPEG, achieves better coding performance than other existing video coding standards [2, 3]. The improvement in coding efficiency stems mainly from the prediction part such as motion estimation with quarter-pixel accuracy, multiple reference frames, spatial intra-prediction and variable block sizes and various prediction modes. Since a variety of block sizes and prediction modes are supported, the rate–distortion optimisation (RDO) technique [3] is employed prior to coding which exhaustively examines all intra- and inter-prediction modes to search the best block size and prediction mode for the current macroblock (MB). This full search fashion achieves considerable coding performance, but causes high computational complexity and energy consumption. For mobile video applications where real-time video coding and streaming over wireless communication networks are performed under energy constraints, not only high image reconstruction quality and low bit rates (BRs) are required, but also a low complexity scheme is desired. Although mobile smart phones are getting more powerful, they are still not competitors to their sibling desktop PCs. Therefore it is always motivational to reduce the computational complexity of mobile tasks. This is not only to reduce task response time, but also to potentially reduce battery consumption to prolong the operational lifetime of mobile devices. Since intra-prediction mode decision is one of the most computationally complex modules in the H.264/AVC encoder, speeding up this process is normally considered a reasonable solution to this task.

For the purpose of reducing coding complexity, many fast intra-prediction mode decision algorithms have been proposed to filter some unlikely prediction modes. Huang *et al.* [4] developed a context-based adaptive algorithm which incorporates a sub-sampled matching criterion to skip unlikely candidate modes, saving about 50% encoding time. Meanwhile, Pan *et al.* [5] proposed an algorithm prepruning some prediction modes according to the distribution of the edge direction histogram established by applying Sobel edge detecting operator. Then, Wang *et al.* [6] suggested a dominant edge strength (DES) mode decision scheme incorporating MPEG-7 edge descriptors into a sub-sample technique to determine the dominant edge to reduce candidate modes. Tsai *et al.* [7] proposed a sub-block-based direction detection and a pixel-based direction detection algorithm by improving MPEG-7 edge descriptors. Later, Tsai *et al.* [8] also proposed another efficient mode decision algorithm removing unlikely modes from candidate mode list by employing the intensity gradient technique to detect the edge orientation in the block. Huang *et al.* [9] improved the DES algorithm [6] and proposed a three-stage approach which takes the correlation between blocks into account and treats the most probable mode (MPM) as the default candidate mode instead of the direct current (DC) mode. In most recent work, Lim *et al.* [10] proposed an inner-8×8 block variance-based block size decision method and a mean deviations-based intra-prediction mode decision algorithm. Pejman and Zargari [11] presented an orthogonal prediction modes filtering-based fast mode decision algorithm. Some other efficient methods were reported in [12–14].

By employing a variety of techniques to detect the edge strength in the current predicting block, numerous fast algorithms achieve a good coding performance-complexity trade-off. However, despite the texture information in the current block being exploited, the correlation between current block and the neighbouring reference pixels used to predict the predicting pixels is not taken into account in most algorithms. The best trade-off between coding performance and computational complexity is still not achieved. Recently, some mode decision algorithms based on machine learning method achieve a good trade-off between computation load and coding performance [15, 16]; however, an additional training process is involved. Thus, these algorithms cannot be employed in real-time applications. Furthermore, for mobile video applications, not only efficient video coding is demanded, but also flexible video coding with different granularities of coding complexity is desired to adapt various power conditions. Most fast mode decision algorithms developed so far, however, do not address this issue. In this paper, we firstly propose an efficient hierarchical fast intra-prediction mode decision algorithm, based on our previous work [17, 18]. And then, a complexity algorithm is introduced to achieve flexible control of energy consumption.

The proposed fast algorithm adopts a hierarchical structure in line with the process of H.264/AVC intra-prediction mode decision which includes block size and mode decision stages. In the block size decision stage, a discretisation total variation (DTV)-based method is employed to eliminate an unlikely block size. And in the mode decision stage, we apply the orientation gradients of directional intra-prediction modes as a filter to prune unlikely directional modes. Since the orientation gradients are computed by the differences between the reference pixels and the predicting pixels in the prediction directions, they measure both the edge strength between the reference pixels and the current block in the prediction directions and prediction residual energies of prediction modes. It can be desired to filter unlikely modes more accurately than those methods based on detecting the texture information inside the current block. In addition, inspired by [9], considering the spatial locality of the prediction mode, we improve the fast algorithm incorporating with the MPM to further improve the coding performance. Furthermore, we observe the fact that the coding performance improves and the computational complexity increases with increasing of the number of the candidate modes participating in RDO. That is, employing

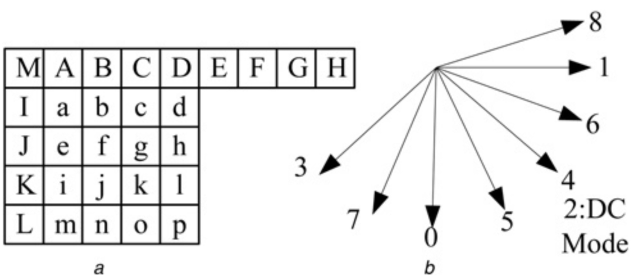


Fig. 1 *Preliminaries of H.264/AVC intra-prediction modes*

a 4 × 4 block with its reference pixels
b Nine I4MB prediction modes

more modes for RDO leads better coding performance, but higher complexity, and vice versa. Therefore we introduce a complexity scalable algorithm to achieve flexible control of energy consumption by adjusting the number of candidate modes. We define five different levels of complexity with different coding performances in the algorithm.

The remainder of this paper is organised as follows. We introduce some preliminaries of H.264/AVC intra-prediction in Section 2. Then, the proposed hierarchical fast mode decision algorithm and the complexity scalable algorithm are presented in Section 3. Experimental results are reported in Section 4. Finally, we conclude this paper in Section 5.

2 Preliminaries of H.264/AVC intra-prediction

Different from some previous video coding standards where intra-prediction is conducted in the transform domain, intra-prediction in H.264/AVC is always conducted in the spatial domain. H.264/AVC codes the frame by 16 × 16MB. For intra-prediction, there are three luma strategies suggested in H.264/AVC high profile: I4MB, I8MB and I16MB. Each of them makes prediction in 4 × 4, 8 × 8 and 16 × 16 block size. Pixels of the current block are predicted by referring to up and/or left blocks previously coded and reconstructed. For example for I4MB, the current block pixels *a*–*p* are predicted by the neighbouring pixels *A*–*M* as illustrated in Fig. 1*a*. The residual between the actual samples and their predicted values is then coded. And as showed in Fig. 1*b*, besides the DC prediction mode, eight directional prediction modes are specified for I4MB. The same nine modes are defined for I8MB. Four prediction modes including the DC mode and three directional modes are supported for I16MB. Each 4 × 4 or 8 × 8 block can utilise different modes to make prediction. However for I16MB for I4MB and I8MB, only a single prediction mode is permitted for an entire MB. The rate–distortion (*R*–*D*) costs of all the sub-blocks in the MB are accumulated as the *R*–*D* cost of the MB. Then, the MB is coded by adopting the optimal block size and prediction mode. For chroma intra-prediction, since only 8 × 8 block size with the same four modes as those defined for I16MB prediction is supported, the mode with the minimum *R*–*D* cost is selected to predict both the *U* and *V* components.

The choice of prediction modes for chroma components is independent to that of luma components. Since there are four chroma intra-prediction modes, for each MB, a total of $(4 \times 4 \times 9 + 2 \times 2 \times 9 + 4) \times 4 = 736$ different *R*–*D* cost calculations are needed to find out the block size and prediction mode to achieve the best coding performance for intra-prediction. The coding performance for each prediction mode is measured by the *R*–*D* cost which is defined as

$$J(\text{mode},\ \text{QP}) = D(\text{mode},\ \text{QP}) + \lambda \times R(\text{mode},\ \text{QP}) \quad (1)$$

where *D* represents the distortion of the reconstructed images which is commonly measured by the sum of square difference between the current block and the reconstructed block. λ is the Lagrange multiplier. *R* denotes the bits of the coded MB, mode denotes the prediction mode. QP represents the quantisation parameter (QP) and *J* represents the *R*–*D* cost of the prediction mode. For every prediction mode, a series of procedures including transformation, quantisation, inverse transformation, inverse quantisation and entropy coding are needed to attain the value of *R*. And for the value of D, same procedures except entropy coding are involved. Therefore the RDO process is very complicated and energy consuming.

3 Proposed algorithms

3.1 DTV-based block size filter

For luma samples intra-prediction, H.264/AVC exhausts all the modes of the three block size to pursue the optimal prediction mode. In practice, smaller block sizes are used more often than larger block sizes for texture regions, whereas larger sizes are more often used for flat regions. The percentage of the best intra MB prediction block size for various sequences is presented in Fig. 2. There exists the tendency that I4MB is the majority mode for the lower resolution sequences in which an MB has more details. With the resolution and the homogeneity of the texture of an MB increasing, the percentage of employing I4MB decreases, whereas the percentage of adopting I16MB increases. In other words, I16MB prediction is unlikely to be the optimal block size for a high detailed MB and I4MB prediction is unlikely the best scheme for a homogeneous MB. Therefore we can determine the coding block size for intra-prediction according to the texture complexity of the MB prior to mode decision to shrink the candidate mode set. That is, the larger block size prediction mode can be eliminated when the complexity of an MB is high, otherwise the smaller block size prediction schemes are skipped. As shown in Fig. 2, the I8MB scheme as an intermediate mode between I4MB and I16MB plays an important role to retain the prediction accuracy. We always reserve I8MB as a candidate scheme in the proposed algorithm.

A texture MB features dense and large-magnitude gradients between the adjacent pixels, whereas a smooth MB means sparse and fine intensity dissimilarities in the MB. The density and magnitude of the dissimilarities between the adjacent pixels can be measured by the total variation (TV) of the MB which is defined as follows

$$\begin{aligned}\mathrm{TV}_{\mathrm{MB}} &= \|\nabla I_{\mathrm{MB}}\|_{\mathrm{TV}}\\ &= \sum_{i=0}^{14}\sum_{j=0}^{14}\sqrt{(I[i,j]-I[i+1,j])^2+(I[i,j]-I[i,j+1])^2}\end{aligned} \tag{2}$$

where $I[i, j]$ is the luma intensity of the pixel at i, j of the MB. The calculation of the TV includes multiplication and square root operations. To simplify the calculation, we use a discretisation of the TV formulated in (3) as a replacement in this paper

$$\begin{aligned}\mathrm{DTV}_{\mathrm{MB}} &= \|\nabla_i I_{\mathrm{MB}}\|_{\mathrm{TV}} + \left\|\nabla_j I_{\mathrm{MB}}\right\|_{\mathrm{TV}}\\ &= \sum_{i=0}^{14}\sum_{j=0}^{14}\left(\left|I[i,j]-I[i+1,j]\right|+\left|I[i,j]-I[i,j+1]\right|\right)\end{aligned} \tag{3}$$

The value of $\mathrm{DTV}_{\mathrm{MB}}$ ranges from 0 to 115, 200. The larger the $\mathrm{DTV}_{\mathrm{MB}}$ is, the denser and more significant the dissimilarities between pixels are and the more complex the MB is. The complexity of an MB is classified according to $\mathrm{DTV}_{\mathrm{MB}}$ as follows

$$\text{complexity} = \begin{cases}\text{high} & \mathrm{DTV}_{\mathrm{MB}} > T\\ \text{low} & \text{otherwise}\end{cases} \tag{4}$$

where T is a threshold. If the value $\mathrm{DTV}_{\mathrm{MB}}$ is larger than the threshold, the I16MB prediction scheme is eliminated, otherwise I4MB is skipped. The hit-rates after eliminating a block size scheme using various thresholds under four QPs is shown in Fig. 3. Here, the hit-rate is defined as the ratio between the number of MBs in which our method chooses the same best mode as the JVT H.264/AVC reference software JM18.0 [19] and the total number of MBs in the test sequences. As indicated in Fig. 3, the threshold around 800 achieves relatively high hit-rates for various resolution sequences under different QPs. We adopt this value for block size filtering in this paper.

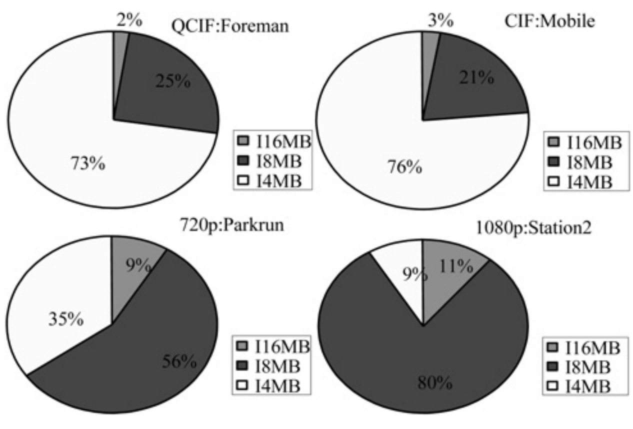


Fig. 2 *Percentage of the best MB prediction scheme*

3.2 Orientation gradient-based mode decision

Making a decision for coding block size in advance eliminates a block size scheme, but the computational load for prediction mode decision is still very high. Some further mode filtering policies are needed, especially for MBs with high complexity. In this section, we present the baseline orientation gradient-based mode decision algorithm.

The orientation gradient of a directional prediction mode is defined as (5)

$$G(\mathrm{dir}) = \sum_{X\in R, x\in C, Xx//\mathrm{dir}} \left|P_X - I_x\right|/n \tag{5}$$

where dir denotes prediction direction. R and C are subsets sampled from the reference pixels and the current block pixels, respectively, and n is the number of sampled pixels. Xx//dir represents the line connecting the reference pixel X and the current block pixel x is in the direction dir. P_X is the reconstructed value of the reference pixel X belonging to R. I_x represents the original value of the current block pixel x belonging to C.

G (dir) in (5) characterises the gradient between the reference pixels and the current block whose value reflects the edge strength of the orthogonal direction of the prediction mode. On the other hand, the orientation gradient of a prediction mode can be regarded as an approximation of the absolute average of the prediction residual of the prediction mode. It is reasonable to adopt the orientation gradient as a prediction mode coding performance estimation model to filter some unlikely modes prior to R–D cost calculation.

IET Commun., 2014, Vol. 8, Iss. 9, pp. 1654–1662
doi: 10.1049/iet-com.2013.0797

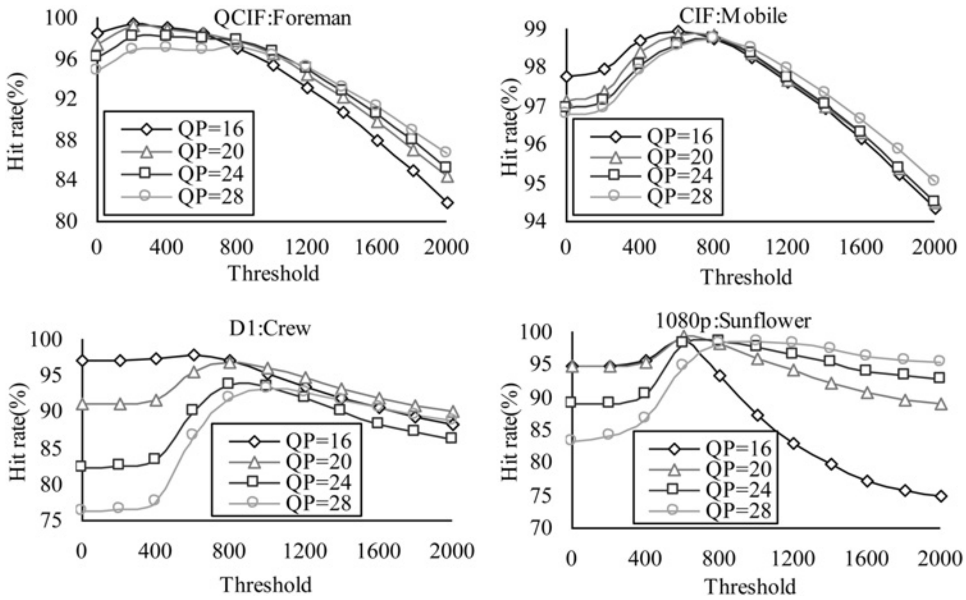


Fig. 3 *Hit-rates of different DTV thresholds for various resolution sequences under four QPs*

In I4MB prediction scheme, the current block pixels *a–p* are predicted by the neighbour pixels *A–M* with nine modes as indicated in Fig. 1. We sample four representative predicting pixels and some reference pixels to calculate orientation gradient corresponding to each directional prediction mode as showed in Fig. 4. For example, following Fig. 4*a*, the orientation gradient of vertical prediction mode is calculated by (6)

$$G(A) = |P_A - I_e| + |P_A - I_m|$$
$$G(C) = |P_C - I_g| + |P_C - I_o| \quad (6)$$
$$G(\text{ver}) = (G(A) + G(C))/4$$

where P_A and P_C denote the reconstructed values of the reference pixels *A* and *C*, respectively. I_e, I_m, I_g and I_o represent the original values of the current block pixels *e*, *m*, *g* and *o*, respectively. The lines *Ae*, *Am*, *Cg* and *Co* are in the vertical prediction direction as indicated in Fig. 4*a*. *G*(*A*) and *G*(*C*) represent the gradients between the reference pixels *A* and *C* and the current block in the vertical direction. *G*(ver) denotes the orientation gradient of the vertical mode which is also an estimation of predictions residuals of the current block. Following Figs. 4*b–h*, the orientation gradients of the other seven directional modes for I4MB can be obtained.

The same way adopted in I4MB is employed to calculate the orientation gradients of the directional modes for I8MB, whereas eight predicting pixels are sampled. For I16MB, we sample sixteen representative predicting pixels and relevant reference pixels in each prediction direction for the calculations of the orientation gradients.

If the orientation gradient of a directional prediction mode is small enough or smaller than other orientation gradients, it means no or tiny difference between the reference pixels and the current block pixels in the prediction direction. Therefore this prediction mode may be a suitable mode for current block prediction. Otherwise, there should be a strong edge in the orthogonal direction of the prediction direction, and the prediction mode is unlikely to be the best mode for prediction. The mode with smallest orientation gradient is

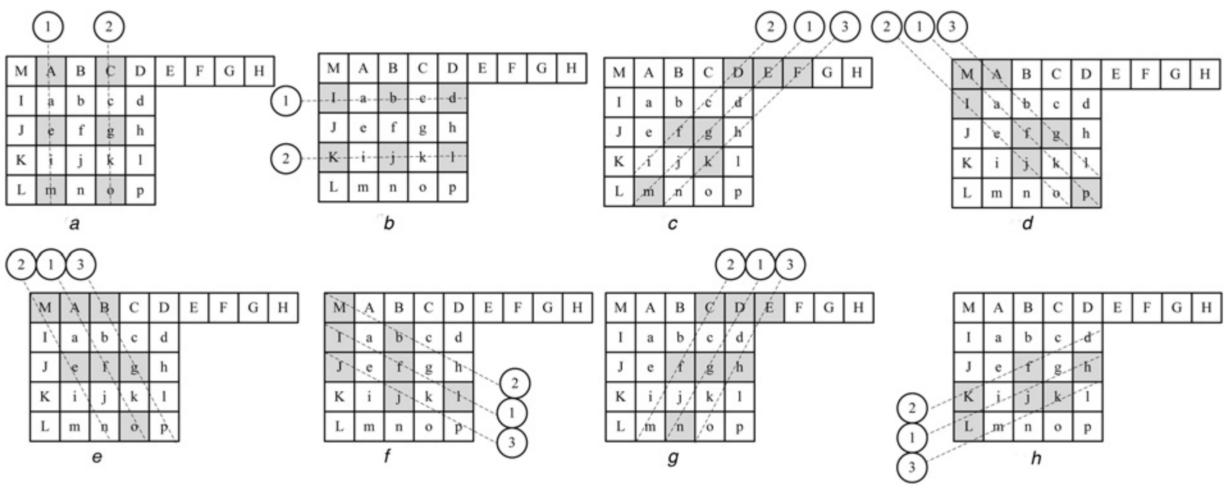


Fig. 4 *Orientation gradients of the directional modes for I4MB, (a) and (b) responding to mode 0, 1, 3, …, 8*

a Mode 0
b Mode 1
c Mode 3
d Mode 4
e Mode 5
f Mode 6
g Mode 7
h Mode 8

Algorithm 1:

Input: *Level* – 0-4, user input

I[16,16] – values of pixels in current MB

P – values of reference pixels

T – threshold for block size filtering

Output: the best block size and prediction modes for MB

if *Level*>0 then //perform block sizes and modes filtering

$$DTV_{MB} \leftarrow \sum_{i=0}^{14}\sum_{j=0}^{14}(|I[i,j]-I[i+1,j]|+|I[i,j]-I[i,j+1]|).$$

if DTV_{MB}>*T* then

Candidate_BlockSize_Set←{*I4MB,I8MB*}

else

Candidate_BlockSize_Set←{*I8MB, I16MB* }

end if

if *Level*>1 then

determine the number of candidate modes according to *level*

$$G(dir) \leftarrow \sum_{X\in R, x\in C, Xx//dir} |P_X - I_x| \Big/ n$$

remove modes with larger *G(dir)* from candidate modes

if MPM not in candidate modes //only for I4MB and I8MB

replace the DC mode in candidate modes with MPM

endif

endif

else //perform full search

Candidate_BlockSize_Set←{*I4MB,I8MB, I16MB* }

end if

perform RDO in the candidate modes

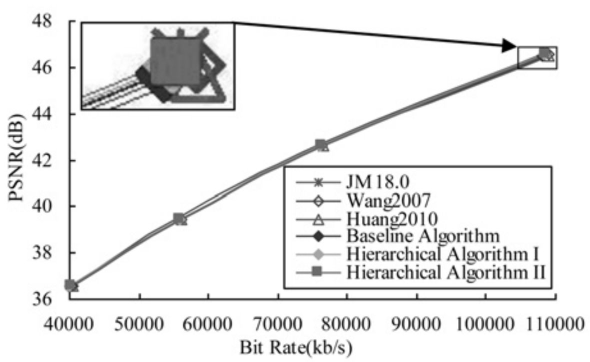
Fig. 5 *Complexity scalable mode decision algorithm*

more likely to be the best mode than other modes, but not necessarily the best model for the current block in the actual encoding process. To preserve coding performance, we cannot utilise the mode with the smallest orientation gradient as the only prediction mode. Moreover, the DC mode (mode 2) for all the block size schemes is always reserved as a supplement to retain coding performance in a smoother block or a boundary block. Thus, in practice, five directional modes with larger orientation gradients are filtered and three with smaller orientation gradients along with the DC mode are reserved to perform RDO in I4MB and I8MB. For I16MB, since only three directional modes are involved, we only reserve the directional mode with the smallest orientation gradient along with the DC mode for RDO.

Chroma intra-prediction which supports only 8×8 block size with the same four prediction modes as those of I16MB has much lower computational load, so we leave it out in this paper. Therefore no performance loss is introduced in chroma components by our algorithm.

3.3 Improved hierarchical block size and mode decision algorithm

The block size and mode decision method described above forms an efficient two-stage hierarchical block size and mode decision algorithm for intra-frame coding. The algorithm exploits contextual information of the current block adequately, yet it does not take the spatial correlation of the prediction modes into account which limits the prediction accuracy. In natural images, there exists the same texture direction between the neighbouring blocks in many regions, thus, the prediction modes adopted by the adjacent blocks are likely the best mode for the current block. The MPM for I4MB and I8MB is defined as the prediction mode adopted in the left or the upper neighbour with the smaller prediction mode number. If neither of these two neighbours is available, the default DC mode is used. Treating the MPM as one of the candidate modes is expected to increase the coding performance of the fast algorithm. Inspired by [9], we improve the proposed block size decision algorithm by always including the MPM as a candidate mode for I4MB and I8MB predictions instead of the DC mode. If the MPM is one of the three candidate directional modes, the DC mode is added into the candidate modes list. For I16MB, as the MPM is not defined, the directional mode with the smallest orientation gradient along with the DC mode is reserved for RDO. Since still four prediction modes for I4MB and I8MB and two modes for I16MB are examined, the improved algorithm incorporating with the MPM is expected to increase the coding performance with the same computational load as the baseline algorithm.



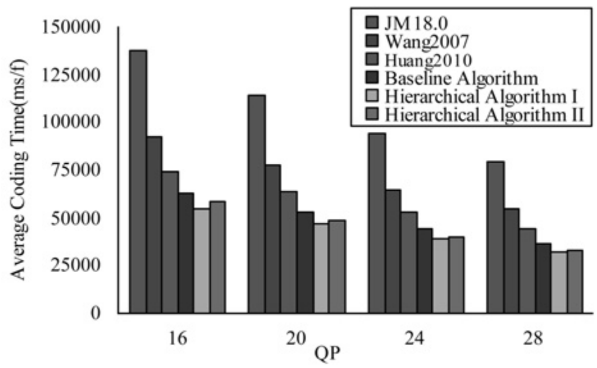


Fig. 6 *Performances comparison among different fast algorithms for Parkrun (720p)*

a R–D curves

b Encoding time

doi: 10.1049/iet-com.2013.0797

3.4 Complexity scalable mode decision algorithm

In the proposed hierarchical algorithm, fixed directional modes are reserved for RDO, thus the computational complexity of the algorithm is fixed. In fact, performing RDO with different numbers of modes will lead to different computational load. Reserving more modes means better coding performance, but more computational complexity. Adopting fewer modes reduces more coding performance, but also reduces more computational complexity. Motivated by this fact, we introduce a parameter to control the number of modes participating in RDO and adjust the coding complexity. To streamline processing, we measure coding complexity on five levels in this paper. Level 0 performs full search, and level 1 only performs block size filtering. Level 2 to level 4 perform block size and mode filtering. And 6, 4 and 2 modes of I4MB and I8MB with smaller orientation gradients are reserved for RDO in levels 2–4, respectively. The directional mode with the smallest orientation along with the DC mode of I16MB is always retained for levels 2–4. The algorithm can be summarised as Algorithm 1 (see Fig. 5).

4 Experimental results and analysis

4.1 Performance evaluation for mode decision algorithms

In this section, we evaluate the performance of the proposed fast mode decision algorithm in terms of the peak signal-to-noise ratio (PSNR) loss, the BR increase and the complexity reduction. Ten different resolution sequences are chosen for testing with all I-frame coding under different QPs 16, 20, 24 and 28 in high profile. A total of 300 frames are encoded and RDO and CABAC entropy coding is enabled. The frame rate is set to 30. The performance gain or loss is measured with respect to the JM full search scheme, represented by ΔP and ΔBR. They are calculated by the Bjontegaard's method [20] across the given four QPs. The complexity can be measured by either the encoding timesaving or the number of instructions required. Since a series of complex procedures including transformation, quantisation, inverse transformation, inverse quantisation and entropy coding are involved in the RDO process, it is hard to count the vast number of instructions required statically. To simplify the measurements, the encoding timesaving is adopted in this paper. It is obtained by (7)

$$\Delta T = \frac{T_{\mathrm{JM}} - T_{\mathrm{method}}}{T_{\mathrm{JM}}} \times 100\% \qquad (7)$$

where T_{method} denotes the average encoding time of the fast method across various QPs. T_{JM} represents the average encoding time of JM. And ΔT is the average encoding timesaving of the fast method with respect to JM.

Besides JM full search scheme, two typical fast mode decision algorithms, Wang2007 [6] and Huang2010 [9], are chosen as the benchmarks. Since Wang2007 did not provide any mode decision scheme for the 8×8 luma block, we treat it the same as the 4×4 luma block with just the different 2×2 pseudo-block size. Three approaches are involved to evaluate the efficiency of the proposed methods: (i) the orientation gradient-based baseline mode decision algorithm (named as baseline algorithm), (ii) the variance-based block size filter adopted in Huang2010 and

Table 1 Performance gains/losses of fast algorithms for various sequences

Sequences	Wang 2007			Huang 2010			Baseline algorithm			Hierarchical algorithm I			Hierarchical algorithm II		
ΔP, dB	ΔBR,%	ΔT,%	ΔP,dB	ΔBR,%	ΔT%	ΔP,dB	ΔBR,%	ΔT,%	ΔP, dB	ΔBR,%	ΔT,%	ΔP, dB	ΔBR,%	ΔT,%	
Foreman(QCIF)	−0.155	1.742	31.93	−0.145	1.643	39.44	−0.110	1.229	53.91	−0.090	1.021	57.29	−0.091	1.02	57.64
News(QCIF)	−0.291	2.721	37.58	−0.242	2.280	45.14	−0.171	1.585	54.12	−0.172	1.597	57.82	−0.149	1.395	54.95
Paris(CIF)	−0.202	1.838	36.41	−0.212	1.944	44.43	−0.133	1.206	53.87	−0.139	1.265	58.40	−0.119	1.395	56.37
Mobile(CIF)	−0.189	1.375	29.4	−0.168	1.225	32.78	−0.139	1.010	53.8	−0.135	0.987	55.87	−0.143	1.051	59.14
Crew(D1)	−0.105	2.105	36.91	−0.064	1.284	66.17	−0.084	1.692	53.25	−0.083	1.676	69.15	−0.058	1.161	60.49
Soccer(D1)	−0.137	1.403	34.13	−0.100	1.014	54.14	−0.103	1.048	53.41	−0.100	1.018	60.51	−0.116	1.18	58.76
Shields(720p)	−0.095	1.430	32.51	−0.102	1.518	50.17	−0.066	0.975	53.47	−0.068	0.998	61.77	−0.079	1.175	54.42
Parkrun(720p)	−0.116	1.156	32.23	−0.140	1.397	45.32	−0.083	0.829	53.79	−0.088	0.878	59.18	−0.098	0.982	57.65
Station2(1080p)	−0.104	2.645	38.27	−0.079	1.926	67.98	−0.073	1.821	53.17	−0.082	1.987	70.81	−0.067	1.604	69.68
Sunflower(1080p)	−0.073	2.605	43.87	−0.057	2.034	71.27	−0.068	2.406	53.04	−0.069	2.418	72.56	−0.06	2.109	74.99
average	−0.147	1.902	35.32	−0.131	1.627	51.68	−0.103	1.380	53.58	−0.103	1.384	62.34	-0.098	1.307	60.41

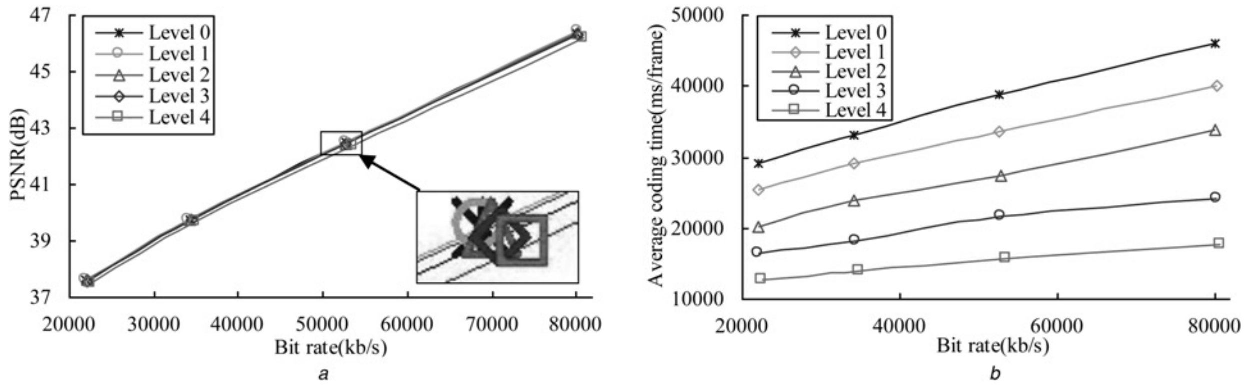


Fig. 7 *R–D and R–C curves for different levels for Shields(720p)*
a R–D curves
b R–C curves

the proposed orientation gradient-based mode decision integrated hierarchical algorithm (named as hierarchical algorithm I) and (iii) the novel DTV and orientation gradient-based hierarchical algorithm (named as hierarchical algorithm II). All the algorithms are implemented on JM 18.0 [19]. The baseline algorithm treats the DC mode as the default mode to compare with Wang2007, and the hierarchical algorithm I and hierarchical algorithm II are both improved by replacing the DC mode as the MPM. Note that, the value of 40 000 for the threshold of the variance-based block size filter of the hierarchical algorithm I is adopted, different from the value of 90 000 implied in Huang2010.

Fig. 6 presents the *R–D* curve and encoding time comparison for Parkrun(720p) between JM18.0 full search fashion and the fast algorithms. It can be observed that the hierarchical algorithms I and II achieve significantly better results than other algorithms in reducing the encoding time with negligible coding performance loss relative to JM18.0. In fact, the similar results are achieved for other sequences with various resolutions. Detailed comparative results on ten different resolution test sequences are summarised in Table 1. It can be observed that the baseline algorithm achieves more encoding timesaving and better coding performance than Wang2007's algorithm. Huang2010's algorithm improves Wang2007's in terms of both the coding performance and, more notably, the computational complexity, but the hierarchical algorithms I and II outperform both in terms of PSNR, BR and encoding timesaving. These imply the orientation gradient employed in the proposed algorithm detects unlike modes more accurately than the texture information in the block applied in most other fast algorithms. It is also shown in Table 1 that the hierarchical algorithm II employing the novel DTV-based block size filter achieves a perceivable better coding performance than the hierarchical algorithm I adopting the variance-based block size filter with only a near 2% timesaving reduction. The proposed DTV-based hierarchical algorithm even achieves a better coding efficiency than the baseline algorithm with a significant improvement in timesaving. This indicates the proposed DTV-based method filters the unlike block size more accurately than the variance-based method. And note that, as depicted in Table 1, the proposed algorithm performs better for high-resolution sequences which are getting more common both in desktop and mobile video applications. This trend is reasonable because there exist more homogeneous regions in high-resolution sequences and the DTV-based block size filter chooses I16MB for coding more frequently and accurately which leads both the coding efficiency and the timesaving improvement.

4.2 Performance evaluation for complexity scalable algorithm

In this section, we evaluate the performances of the proposed complexity scalable algorithm in various levels in terms of the degradation of the coding performance and the reduction of the computational complexity. The same experimental conditions and measurements described in the above section are employed.

We firstly investigate the *R–D* curves and the *R–C* curves for different complexity levels of the Shields(720p) sequence which are plotted in Fig. 7. As illustrated in Fig. 7*a*, the coding performance of the proposed algorithm is successively decreasing from level 0 to level 4. However under the same conditions, both the degradation of PSNR and the increase of BR from level 0 to level 3 are inconspicuous. The coding performance degradation of level 4 with respect to other levels is relatively observable, but still in the tolerable range. On the other hand, as shown Fig. 7*b*, the complexity of the algorithm is successively reducing from level 0 to level 4. With the same QP, the reduction of the complexity across levels is remarkable, and about 20% encoding time is saved in the coding process from one level to the next.

Gains or losses of the coding performance and the complexity of levels 1–4 relative to level 0, that is, JM full search fashion, for ten test sequences are listed in Table 2 in detail. It indicates that the same observation existing in Shields(720p) sequence holds for other sequences with various resolutions too. For level 1 and level 2, the algorithm achieves average 18.14 and 39.64% complexity reduction relative to JM full search scheme with an imperceptible loss of the coding performance, respectively. They can be employed under moderate energy conditions. Level 3 corresponds to the proposed hierarchical fast algorithm which achieves a better performance-complexity trade-off. And level 4 achieves average 73.03% complexity reduction with a relatively high but acceptable degradation of the coding efficiency. It can be applied under low energy conditions.

IET Commun., 2014, Vol. 8, Iss. 9, pp. 1654–1662
doi: 10.1049/iet-com.2013.0797

Table 2 Performance gains/losses of level 1–level 4 relative to JM18.0 for various sequences

Sequences	Level 1			Level 2			Level 3			Level 4		
ΔP, dB	ΔBR, %	ΔT, %	ΔP, dB	ΔBR, %	ΔT, %	ΔP, dB	ΔBR, %	ΔT, %	ΔP, dB	ΔBR, %	ΔT, %	
Foreman(QCIF)	−0.005	0.052	14.99	−0.037	0.414	37.82	−0.091	1.02	57.64	−0.353	4.187	67.98
News(QCIF)	−0.017	0.154	6.84	−0.066	0.61	29.47	−0.149	1.395	54.95	−0.365	3.456	68.82
Paris(CIF)	−0.045	0.149	5.64	−0.062	0.564	29.45	−0.119	1.395	56.37	−0.375	3.060	69.87
Mobile(CIF)	−0.005	0.032	8.45	−0.06	0.433	35.83	−0.143	1.051	59.14	−0.355	3.380	70.13
Crew(D1)	−0.004	0.09	13.00	−0.028	0.567	46.13	−0.058	1.161	60.49	−0.152	3.076	75.45
Soccer(D1)	0.000	−0.003	11.63	−0.047	0.482	33.51	−0.116	1.18	58.76	−0.3	3.100	71.53
Shields(720p)	−0.003	0.056	12.42	−0.037	0.534	32.60	−0.079	1.175	54.42	−0.233	3.472	72.94
Parkrun(720p)	0.004	−0.034	17.84	−0.039	0.394	34.60	−0.098	0.982	57.65	−0.296	2.986	72.30
Station2(1080p)	−0.008	0.065	34.74	−0.033	0.723	51.34	−0.067	1.604	69.68	−0.183	4.705	77.62
Sunflower(1080p)	−0.007	0.21	55.81	−0.03	1.047	65.61	−0.06	2.109	74.99	−0.169	6.161	83.66
average	−0.009	0.0771	18.14	−0.044	0.577	39.64	−0.098	1.3072	60.41	−0.278	3.758	73.03

5 Conclusion

In this paper, we firstly present a hierarchical block size and mode decision algorithm for H.264/AVC intra-prediction video coding. By skipping an unlikely block size scheme according to the DTV of the current MB and some directional modes with larger orientation gradients prior to RDO, the encoding complexity of the proposed hierarchical fast block size and mode decision algorithm is decreased significantly with a negligible coding performance degradation. Incorporating with MPM, the code efficiency obtains further improvement. Experimental results demonstrate the proposed fast hierarchical algorithm achieves better performance-complexity trade-off than the state-of-the-art fast mode decision algorithms. Furthermore, to achieve flexible control of energy consumption, we introduce a complexity scalable algorithm based on the proposed fast algorithm. We measure the coding complexity on levels 0–4 according to the number of modes reserved for RDO. Users can employ an appropriate complexity level for video coding according to power conditions. For future research, since the intra-prediction process in the new high-efficiency video coding (HEVC) has a similar structure with it in H.264/AVC, one of our works is to extend our fast mode decision algorithm and complexity scalable algorithm to HEVC.

6 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (no. 61072122), Hunan Province Science and Technology Planning Project (nos. 2012FJ4129, 2012GK3056 and 2013FJ4033), Changsha Science and Technology Planning Key Project (nos. K1104022-11 and K1207027-11) the Science Research Key Project of the Education Department of Hunan Province (no. 13A107), UK EPSRC Project DANCER (EP/K002643/1), EU FP7 Projects MONICA (GA-2011-295222), and CLIMBER (GA-2012-318939).

7 References

1 Draft ITU-T recommendation and final draft international standard of joint video specification, ITU-T rec. H.264|ISO/IEC 14496-10 AVC, 2003
2 Report of the Formal Verification Tests on AVC, ISO/IEC 14496-10 ITU-T Rec. H.264, MPEG2003/N6231, 2003
3 Wiegand, T., Sullivan, G.J., Bjontegaard, G., *et al.*: 'Overview of the H.264/AVC video coding standard', *IEEE Trans. Circuits Syst. Video Technol.*, 2003, **13**, (7), pp. 560–576
4 Huang, Y.W., Hsieh, B.Y., Chen, T.C., *et al.*: 'Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder', *IEEE Trans. Circuits Syst. Video Technol.*, 2005, **15**, (3), pp. 378–401
5 Pan, F., Lin, X., Rahardja, S., *et al.*: 'Fast mode decision algorithm for intraprediction in H.264/AVC video coding', *IEEE Trans. Circuits Syst. Video Technol.*, 2005, **15**, (7), pp. 813–822
6 Wang, J.C., Wang, J.F., Yang, J.F., *et al.*: 'A fast mode decision algorithm and its VLSI design for H.264/AVC intra-prediction', *IEEE Trans. Circuits Syst. Video Technol.*, 2007, **17**, (10), pp. 1414–1422
7 Tsai, A.C., Wang, J.F., Yang, J.F., *et al.*: 'Effective subblock-based and pixel-based fast direction detections for H.264 intraprediction', *IEEE Trans. Circuit Syst. Video Technol.*, 2008, **18**, (7), pp. 975–982
8 Tsai, A.C., Paul, A., Wang, J.C., *et al.*: 'Intensity gradient technique for efficient intra-prediction in .264/AVC', *IEEE Trans. Circuit Syst. Video Technol.*, 2008, **18**, (5), pp. 694–698
9 Huang, Y.H., Ou, T.S., Chen, H.H.: 'Fast decision of block size, prediction mode, and intra block for H.264 intra prediction', *IEEE Trans. Circuit Syst. Video Technol.*, 2010, **20**, (8), pp. 1122–1132
10 Lim, K., Kim, S., Lee, J., *et al.*: 'Fast block size and mode decision algorithm for intra prediction in H. 264/AVC', *IEEE Trans. Consum. Electron.*, 2012, **58**, (2), pp. 654–660

11 Pejman, H., Zargari, F.: 'An efficient fast intra mode decision method based on orthogonal modes elimination', *IEEE Trans. Consum. Electron.*, 2012, **58**, (4), pp. 1445–1452
12 Wang, P., Huang, H., Tan, Z.: 'A fast two-step block type decision algorithm for intra prediction in H. 264/AVC high profile', *Multimedia Tools Appl.*, 2012, **60**, (1), pp. 139–160
13 Chen, C., Chen, J., Ouyang, K., *et al.*: 'A hybrid fast mode decision method for H. 264/AVC intra prediction', *Multimedia Tools Appl.*, 2013, **62**, (3), pp. 719–731
14 Lim, S., Kim, H., Choi, Y., *et al.*: 'Fast intra-mode decision method based on DCT coefficients for H. 264/AVC', *Signal Image Video Process.*, 2013, **7**, (5), pp. 1–9
15 Chiang, C.K., Pan, W.H., Hwang, C., *et al.*: 'Fast H.264 encoding based on statistical learning', *IEEE Trans. Circuit Syst. Video Technol.*, 2011, **21**, (9), pp. 1304–1315
16 Sung, Y.H., Wang, J.C.: 'Fast mode decision for H.264/AVC based on rate-distortion clustering', *IEEE Trans. Multimedia*, 2012, **14**, (3), pp. 693–702
17 Song, Y., Shen, Y.F., Long, J.Z., *et al.*: 'An efficient fast mode decision algorithm for H.264/AVC intra-prediction'. Proc. 11th IEEE Int. Conf. Signal Processing, Beijing, October 2012, vol. 2, pp. 1034–1037
18 Song, Y., Shen, Y.F., Long, J.Z., *et al.*: 'Intra-prediction mode decision algorithm based on orientation gradient for H.264', *Chin. J. Comput.*, 2013, **8**, pp. 1757–1764
19 JVT H.264 /AVC reference software version JM18.0, Available at http://www.iphome.hhi.de/suehring/tml/download/
20 Bjontegaard, G.: 'Calculation of average PSNR differences between RD curves'. Presented at the 13th VCEG-M33 Meeting, Austin, TX, April 2001

IET Commun., 2014, Vol. 8, Iss. 9, pp. 1654–1662
doi: 10.1049/iet-com.2013.0797