Fast Intra- and Inter-Prediction Mode Decision in H.264 Advanced Video Coding

Mehdi Jafari
Islamic Azad University, S & R Branch
Department of Communication Engineering
P.O.Box 14515-775, Tehran, Iran
mjafari@mail.uk.ac.ir

Shohreh Kasaei
Sharif University of Technology
Department of Computer Engineering
P.O.Box 11365-9517, Tehran, Iran
skasaei@sharif.edu

Abstract- H.264/AVC, the latest video coding standard, achieves better video compression rates since it supports new features such as a large number of intra- and inter-prediction candidate modes. H.264/AVC adopts rate-distortion optimization (RDO) technique to obtain the best intra- and inter-prediction, while maximizing visual quality and minimizing the required bit rate. However, full RD cost calculation for all intra-prediction modes and exhaustive searches for optimal motion vectors for all block sizes increase computational complexity considerably with the number of prediction modes allowed. In order to reduce the complexity, we propose an enhanced fast intra-prediction mode selection strategy based on the statistical properties of references pixels and a fast inter-prediction mode decision approach based on the correlation of motion vectors and split/merge procedure. The objective is to reduce the number of candidate modes in intra- and inter-prediction while maintaining the coding efficiency. Experimental results show that our algorithm, compared to the RDO and other fast algorithms, reduces the total encoding time with negligible loss in PSNR and a slightly increased bitrate.

Keywords- H.264/AVC, RDO, intra-prediction, inter-prediction.

1. INTRODUCTION

As recent multimedia applications (using various types of networks) are growing rapidly, video compression requires higher performance as well as new features. The newest video coding standard is developed by the joint of video teams of ISO/IEC MPEG and ITU_T VCEG as the international standard 14496-10 (MPEG-4 part 10) *advanced video coding* (AVC) [1, 2]. H.264/AVC has gained more and more attention; mainly due to its high coding efficiency (the average bitrate saving up to 50% as compared to H.263+ and MPEG-4 Simple Profile), minor increase in decoder complexity compared to existing standards, adaptation to delay constraints (the low delay mode), error robustness, and network friendliness [1, 2]. Table I [3] and Figure 1 [4] show the performance comparisons using MPEG-2, MPEG-4 (ASP), and H.264/AVC. To achieve outstanding coding performance, H.264/AVC employs several powerful coding techniques such as 4x4 integer transform, inter-prediction with variable block-size motion compensation, motion vector of quarter-pel accuracy, in-loop deblocking filter, improved entropy coding such as *context-adaptive variable-length coding* (CAVLC) and *content-adaptive binary arithmetic coding* (CABAC), enhanced intra-prediction, multiple reference picture, and the forth. Due to this new features, encoder computational complexity is extremely increased compared to previous standards. This makes H.264/AVC difficult for applications with low computational capabilities (such as mobile devices). Thus until now, the reduction of its complexity is a challenging task in H.264/AVC.

Table I: Average bit-rate reduction compared to prior coding schemes.

Standards	MPEG-4/ASP	H.263/HLP	MPEG-2
H264/AVC	38.62%	48.80%	64.46%
MPEG-4/ASP	---	16.65%	42.95%
H263/HLP	---	---	30.61%

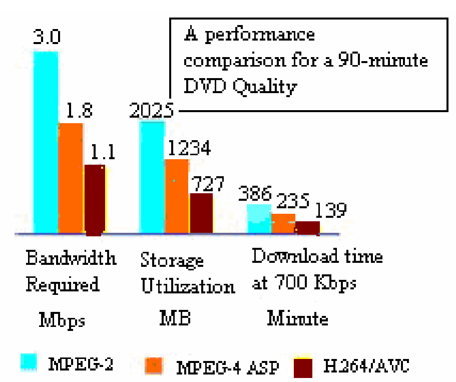


Figure1: Performance comparison of different video coding standards.

Among many new features, the intra- and inter-prediction techniques are recognized to be the main factors that contribute to the success of H.264/AVC. H.264/AVC employs the Lagrangian RDO method to find out the best coding mode of intra-and inter-prediction with highest coding efficiency. Figure 2 [1] shows the RDO process. RDO technique requires a lot of computations since it tests the encoding process with all possible coding modes of intra-coding, and calculates their RD costs to choose the mode having the minimum cost. The reference software of H.264/AVC, JM [15], adopts full search for both motion estimation and intra prediction. The run time percentage of each function is shown in Figure 3[20].

For inter prediction, H.264 supports tree-structured multi-block sizes in motion estimation, permitting seven modes with different block sizes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4). As shown in Figure 3, motion estimation is the most computationally intensive part. Also, The intra-prediction mode decision is very complex and the number of computing RD cost values for luma and chroma of a macroblock is 592 [5].

Therefore, the computational burden of these types of brute force-searching algorithm is far more demanding than any existing video coding algorithm.

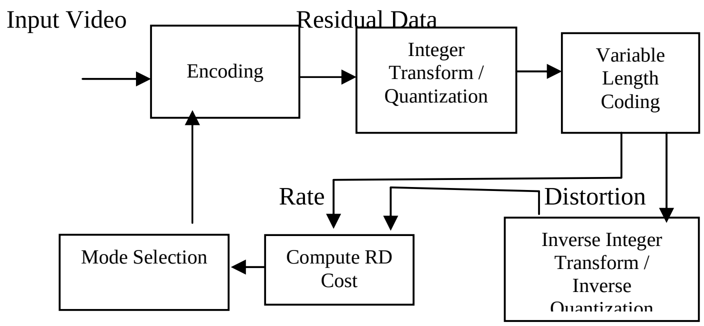


Figure 2: Computation of RD cost [1].

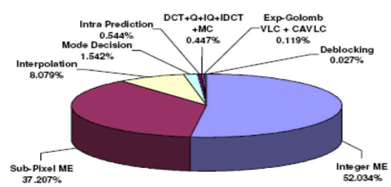


Figure 3: Run time percentages of functional blocks in H.264/AVC baseline encoder [20].

Fast mode selection for intra- and inter-prediction is considered in this paper; which is a challenging subject in H.264/AVC. To reduce the computational complexity, many algorithms (such as fast motion estimation, fast inter-mode prediction, and fast intra-prediction) have been proposed. For fast inter-prediction mode decision, the early termination technique [21] reduces the number of potential prediction modes. In [22], a classification method is proposed to reduce the average number of block types while maintaining the coding performance. This algorithm exploits spatial characteristics of macroblock to help the block segmentation procedure. The FMBME algorithm in [23] is composed of two main steps- early termination based on an adaptive threshold and block segmentation.

For fast intra prediction, since it is a new topic in H.264/AVC coding with respect to other standards such as MPEG-1/2/4 and H.261/H.263 and so far no previous work exists for that. Fast intra-mode decision algorithms using edge detection histogram and local edge detection are proposed in [1, 6, 7]. However, their preprocessing stages still consume a coding time to detect the edge direction and to classify it into a limited direction. The performance of those methods is about 20~30% (or 55~65%) faster than the RDO method at the cost of 2% (or 5%) extra bits. There exist fast algorithms to select the optimal intra-prediction mode using simple directional masks in [8] with saving time of 70%, and statistical-based methods in [9] with saving time of 45%. Another fast intra-mode decision scheme is proposed in [10], where the encoding speed is approximately 30% faster than that of the RDO method. A new fast intra-prediction algorithm based on macroblock properties (FIPAMP) is presented in [11]. This algorithm can achieve 10% to 40% of computation reduction while maintaining similar PSNR and bit rate performance of H.264/AVC codes. In [12], an efficient intra-prediction (EIP) algorithm based on early termination, selective computation of highly probable modes, and partial computation of the cost function is presented. Also, an improved cost function to improve the coding performance is proposed in [13].

In [14], a fast algorithm based on the local edge information obtained by calculating edge feature parameters, and sub-sampling of matching operations is presented. That method can reduce the encoding time about 26% with less than 1.4% used extra bits and no more than 0.7 dB PSNR is sacrificed.

Here, for intra prediction mode decision, we improve Pan's method [1, 6, 7], that is eliminating the DC mode from the candidates if the direction of block is obvious, otherwise, DC mode is chosen. As an alternative method, we proposed a fast intra method using statistical properties of adjacent MBs and reference pixels with combination of the presented algorithms in Pan's method.

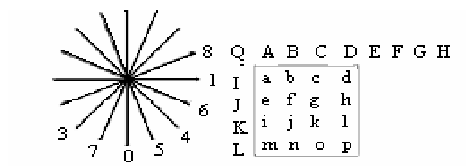
Since the proposed method in this paper analyzes the characteristic of reference pixels, and use the similarity between adjacent MBs, while improved Pan's method analyzes the characteristic of the 4x4 block itself, the combination of these three kinds of methods may achieve better results.

Also, for inter-prediction mode decision, the split/merge procedure is used. In this method a macroblock is divided into quarters of equal areas, then using similarities of motion vectors of adjacent blocks we will show how to merge the sub-blocks for quarter division. Then this split/merge method is used for any 8x8 sub-block. We have verified the proposed algorithm by implementing it on JM7.1 [15] reference software and comparing it with the case of RDO search. Simulation results show the proposed method reduces the encoding time up to 60~75% with loss in PSNR and negligible increase of required bitrate. The remaining parts of the paper are organized as follows. We review the intra- and inter-prediction scheme of H.264/AVC in Section 2. Section 3 presents fast intra- and inter-prediction mode decision algorithms. Simulation results are given in Section 4 and finally Section 5 concludes the paper.

2. INTRA- AND INTER-PREDICTION IN H.264/AVC

2.1 Intra-prediction

H.264/AVC defines a block-based hybrid video codec. In common with earlier standards, H.264/AVC does not define the encoder, but defines the syntax of an encoded video bitstream together with the method of decoding the bitstream [16, 17]. The codec combines intra-picture prediction with inter-picture prediction to exploit the spatial and temporal redundancy, respectively. Intra-prediction is based on the observation that adjacent macroblocks tend to have similar properties. Therefore, as a first step in the encoding process for a given macroblock, one may predict the macroblock of interest from the surrounding macroblocks. The difference between the actual macroblock and its prediction is then coded; which results in fewer bits to represent the macroblock of interest. Prediction may be formed for each 4x4 luma block (I4MB), 16x16 luma MB (I16MB), and 8x8 chroma block. For prediction of 4x4 luminance blocks, the 9 directional modes consist of a DC prediction (Mode 2) and 8 directional modes; labeled 0, 1, 3, 4, 5, 6, 7, and 8 as shown in Figure 4(a). In Figure 4(b), the block (values of pixels "a" to "p") is to be predicted using A to Q. Note that pixels "A" to "Q" from neighboring blocks have already been encoded and may be used for prediction.



(a) (b)

Figure 4: (a) Intra-prediction modes for 4x4 luminance blocks. (b) Labeling of prediction samples.

For regions with less spatial details *(i.e.,* flat regions), H.264/AVC supports 16x16 intra-coding; in which one of four prediction modes (DC, vertical, horizontal and planar) is chosen for the prediction of the entire luminance component of the macroblock as shown in figure 5[18].

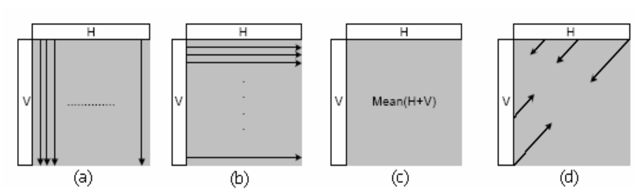


Figure 5: Intra 16x16 prediction modes: (a) Mode 0 (vertical). (b) Mode 1 (horizontal). (c) Mode 2 (DC). (d) Mode 3 (plane) [18].

H.264/AVC supports four chroma prediction modes for 8x8 chrominance blocks, similar to that of the I16MB prediction, except that the order of mode numbers is different: DC (Mode 0), horizontal (Mode 1), vertical (Mode 2), and plane (Mode 3). The same prediction mode is always applied to both chroma blocks. The chroma prediction is independent from luma prediction.

2.2 RDO Procedure for Intra-predction

The RDO procedure to encode one MB in an I-frame is given below [5].

a) Search the best intra-mode for a 4x4 luma block among 9 modes that produces the minimum rate-distortion cost given by:

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

Where *QP* is the macroblock quantization parameter, $\lambda = 0.85 \times 2^{(QP-12)/3}$ is the Lagrangian multiplier, and *MODE* indicates one of the 9 prediction modes of a 4x4 luma block. *R (.)* represents the rate, *i.e.*, the number of bits associated with chosen *MODE*. *SSD (.)* denotes the *sum of the squared differences* between the original 4x4 block luminance signal denoted by *s* and its reconstructed signal denoted by *c*, computed as:

$$SSD(s,c,MODE \mid QP) = \sum_{x=1,y=1}^{4,4} (s(x,y) - c(x,y,MODE \mid QP))^2 \tag{2}$$

b) As contrary to the RDO technique for intra 4x4 luma block mode decision, determine the best intra-mode for a 16x16 macroblock among 4 modes by choosing the mode those results in the minimum s*um of absolute transformed difference* (SATD) given by:

$$SATD = \sum_{(x,y)\in b_k} |T\{I(x,y) - P(x,y)\}| \quad (3)$$

Where I and P represent the true and predicted pixel values, respectively, and T denotes the Hadamard transform.
c) Compare the RD cost for the two best modes, *i.e.*, the I4MB mode obtained from Step 1 and the I16MB mode obtained from Step 2, and choose the best one as the macroblock prediction mode.
d) Determine the best intra-mode for 8x8 chroma block among 4 modes, as I16, by minimizing Equation (1).

2.3 Inter Prediction in H.264/AVC Encoder

As presented in the previous chapter, for inter prediction, there are 7 different block-sizes: 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4, that can be used in inter-frame motion estimation/compensation. These different block sizes actually form a one or two level hierarchy inside a macroblock. Comprising only the first level, the block size can be 16x16, 16x8, or 8x16. In the case of two levels, the marcoblock is specified as P8x8 type, of which each 8x8 block can be one of the subtypes such as 8x8, 8x4, 4x8 or 4x4. The four macroblock type sizes and four macroblock subtype sizes are shown in Figure 6.

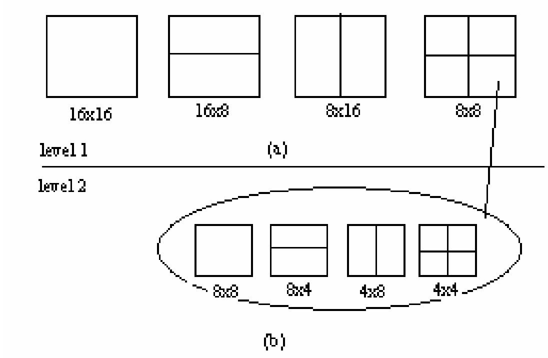
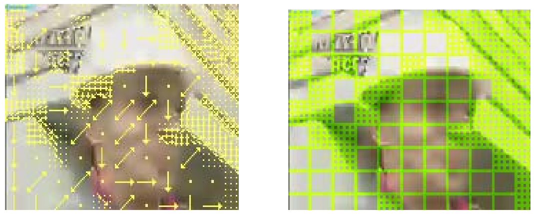


Figure 6: Variable block size in H.264, (a) sizes for a MB type, (b) sizes for a sub MB type.

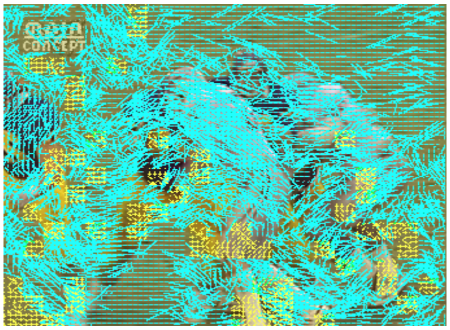
Currently, by trying all the possible block sizes, motion estimation and RDO are performed to find the best block sizes in the macroblock, resulting in very heavy computational load at the encoder.

Finally, for I-frames, while all MBs are predicted as Intra, H.264/AVC encoder encodes the best mode using all mode combinations of luma and chroma and chooses the one that gives the best RDO performance. For P-frames, the encoder performs all possible combination of modes such as variable block sizes used for intra-frame spatial prediction, inter-frame motion estimation, multiple reference frames in the case of inter modes and choose the mode combination that has the lowest RDO cost. Here, two sequences of QCIF and CIF formatted frames are encode using JM7.1. The results are shown for two I- and P-frame in Figures 7 and 8, respectively.



(a) (b)

Figure 7: I-frame of "*foreman*" sequence (QCIF). (a) I4MB and I16MB prediction mode decision.
(b) I4MB and I16MB divisions.



(a)

(b)

Figure 8: P-frame of "*football*" sequence (CIF). (a) I4MB and I16MB (yellow) and inter-prediction (blue) prediction mode decision. (b) I4MB, I16MB, and inter-mode divisions.

3. PROPOSED METHODS FOR FAST INTRA- AND INTER-PREDICTION MODE DECISION

3.1 Fast Intra-Prediction Algorithms

This section presents a new fast intra-prediction algorithm that is based on statistical feature of reference pixels and adjacent block properties. This is motivated by some observations in our experiments. The proposed method is based on several facts that we observed from the statistics of different sequences that:

a) For intra-prediction of luminance samples the probability of 4x4 block size is significantly higher than 16x16 block size at usual quantization parameters (20~35) as shown in figure 9. Therefore fast detection of 4x4 intra-prediction mode can significantly improve the encoding speed at low QP, while 16x16 intra-prediction at large QPs.

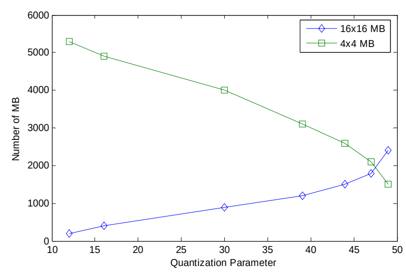


figure 9: Number of 4x4 and 16x16 intra-coded macro blocks at different quantization parameters

b) The prediction modes of each block are correlated with those of neighboring 4x4 luminance blocks. The statistics generated using JM 7.1 encoder [15] shows that for a wide variety of inputs large neighboring blocks have the same I4MB mode.

c) Generally, directional correlation of each block is consistent with directions of the edges[1].

d) There are total 13 reference pixels for intra prediction of a 4x4 luma block, which locate at the up and the left of the 4x4 luma block. Experimental results show that the reference pixels of a 4x4 luma block are similar with each other with a large probability [19].

Based on these observations, we propose a fast intra-prediction mode selection algorithm. In this section some new ideas are combined with the fast mode selection algorithm introduced in [1, 6, 7] to improve their efficiency.

3.1.1 Improved Pan's Method for Fast Decision of I4MB

Pan et al. in [1] present a fast mode selection for intra-prediction method in which the average edge direction of a given block is measured. The Sobel operators are first used to obtain directional vector of each pixel in a block. Then the edge directional histogram of the block is analyzed, and several modes whose direction approximates to the direction of the block are chosen as the candidates. The best combination is searched among the candidates. In Pan's method, for I4MB there are 4 modes (1 DC (mode 2), 1 from maximum amplitude of EDH and its 2 neighbors) while 2 modes (1 DC mode and 1 directional) for each 16x16 luma block and 8x8 chroma block. Here, we improve Pan's method. That is, eliminating the DC mode from the candidates if the direction of the block is obvious, and otherwise, choosing only DC mode. To check whether the DC of the block is clear or not, the *diff* value, given in Equation (1), is checked whether it is smaller than a threshold or not:

$$diff = \sum_{i=0}^{i=15} | avg - p_i | \qquad (4)$$

$$avg = (\sum_{i=0}^{15} p_i + 8) >> 4$$

The improved Pan's method is proposed as follows:

1. For edge directional histogram H, find its maximum. The corresponding mode is denoted by M1.
2. If *diff* > T, RDO procedure is carried out for 3 modes at the most (M1 and its two neighbors).
3. Else, if *diff* <T, RDO procedure I is carried out for two candidate modes at the most. DC with maximum of EDH (M1).
4. For I16MB, based on the same observation as above, after down-sampling by a factor of 2, if *diff1* > T1 only primary prediction mode decided by edge direction histogram is considered as a candidate for the best prediction mode. The *diff1* in this case is presented as:

$$diff1 = \sum_{i=0}^{i=64} | avg - p_i | \quad (5)$$

$$avg = (\sum_{i=0}^{64} p_i + 32) >> 6$$

5. If *diff1* < T1, the maximum prediction mode and DC mode are chosen.
6. For 8x8 chroma block, after down-sampling by a factor of 2, the same procedure as I16MB is used.

Pan's method can reduce RDO calculation from 592 times to 132 times. Here, we improve Pan's method. The number of candidate modes and the RDO calculation in the worst and the best cases are shown in Table II.

Table II. Number of candidate modes.

Block Size	RDO	Pan's Method	Proposed Method (min)	Proposed Method (max)
4x4 (Y)	9	4	2	3
16x16 (Y)	4	2	1	2
8x8 (U/V)	4	3 or 2	1	2

Table II summarizes the number of candidates selected for RDO calculation based on edge direction histogram. As can be seen from Table I, the encoder with the fast mode decision algorithm needs to perform only 33 or 100 RDO calculations, which are much less than that of Pan's method (132) and current H.264 video coding, RDO (592).
.

3.1.2 Fast I4MB Mode Selection Based on Statistical properties of adjacent MBs and Reference Pixels

As an alternative method, we proposed a fast intra method using statistical properties of adjacent MBs and reference pixels with combination of the presented algorithms in previous sections. The method proposed in this section analyzes the characteristic of reference pixels, and use the similarity between adjacent MBs, while improved Pan's method analyzes the characteristic of the 4x4 block itself. So the combination of these three kinds of methods may achieve better results. The proposed algorithm is as follow:

1. For edge directional histogram H, find its maximum. The corresponding mode is denoted by M1.
2. If the modes for one of top or left blocks are M1, then the M1 is chosen as the best candidate mode for the current block. Go to step 7.
3. For 4x4 luma block, MAD(mean of absolute difference) of its reference pixels is computed, if it is smaller than a threshold, M1 is selected. Go to step 7.

This result is yielded from this fact, that if the similarity of reference pixels of a block is high, the difference between different prediction modes will be very small. For this case, it is not necessary to check all of 9 prediction modes, but only one prediction mode is enough [19].

4. If MADH (mean of absolute difference of horizontal references) is less than a threshold and M1 is a member of set {mode 0, mode 3, mode 7 } then M1 is selected. Go to step 7.
5. Also, if MADV (mean of absolute difference of vertical references) is less than a threshold and M1 is a member set of {mode 1, mode 8} then M1 is selected. Go to step 7.

It is obvious that if the similarity of horizontal reference pixels of a block is high, the difference between prediction results obtained with prediction modes 0, 3 and 7 Will be very small. Also, if the similarity of vertical reference pixels of a block is high, the similarity between modes 1 and 8 is high.

6. If all of above conditions are unsatisfied, The improved Pan's method is used.
7. Terminate.

As such, in the worst case only three different 4x4 intra-mode costs will be evaluated. Also, for I16MB and 8x8 chroma block the same algorithm as improved Pan's method is used.

3.1.3 Early Termination of RDO Calculation

Similar to Pan's method, for increasing the speed of the algorithm we use the early termination of RDO calculations for all proposed algorithms as in [1]. During the intra-coding of any prediction mode, the calculation can be terminated if it can foresee that the current mode will not be the best prediction mode. By early termination of the RDO calculations which is deemed to be suboptimal, a great timesaving can be achieved.

3.2 Fast Inter-Prediction mode decision

Observations on coding features of H.264 and valid assumptions are key to present new algorithms, in this paper. This motivated facts for intra-prediction are:

a) The block with higher motion detail, instead of higher texture detail, can be better coded using smaller block size, while the block with less motion detail can be more efficiently encoded using larger block sizes.
b) It is observed that in natural video sequences, when the video objects move, the different parts of the video objects move in a similar manner. From figure 8, we can see that the homogenous regions are encoded using 16x16 block sizes while nonhomogenous regions are encoded using smaller block sizes due.
c) Another observation is that the background is not homogeneous, but because of the temporal stationary is coded using 16x16 block size.

3.2.1 Fast inter-prediction mode decision using split/merge procedure

The split procedure partitions the MB into variable size block using a quad tree approach. In this method a macro block is divided into quarters of equal area (see fig 10). Then using similarities of motion vectors of adjacent blocks we will show how to merge the sub-blocks for quarter division. The proposed algorithm is summarized as follows:

1- Subtract the current frame by its previous frame, then for any 16x16 MB compute:

$$N_i = number\ of\ \ pixels\ \ belong\ to\ the\ set\ MB\quad i$$
$$N_{im} = number\ of\ nonzero\ pixels\ in\ difference\ \ MB$$

2- If N_{im}/N_i is smaller than or equal to the threshold, choose direct mode as final macro block type.

3. Otherwise, the block will be split into four 8x8 blocks, and a new iteration of block matching is conducted for each of these four descending blocks.
4. If motion vectors of 8x8 subblock are equal or three subblock MV are the same and the forth unequal MV only differ by one quarter pixel distance, choose mode 1(16x16). Terminate.
5. If MV0=MV1 and MV2=MV3 (figure 10), the 8x16 is choosed. Terminate.
6. If MV0=MV2 and MV1=MV3, the 16x8 is choosed. Terminate.
7. repeat the steps 2 to 4, for each 8x8 blocks, except that the subblocks are 4x4.
8. terminate.

MV0	MV1
MV2	MV3

Figure 10: MB division.

4. EXPERIMENTAL RESULTS

Our proposed algorithm was implemented into JM7.1[15], provided by JVT according to the test conditions specified in VCEG-N81 document as listed in Table III [24]. Simulations were carried out on the recommended sequences with various quantization parameters for IPPP… type and I-frame only type. For IPPP… experiments, the total number of frames is 300 for each sequence, and the period of I-frame is 100. The used test platform is Pentium IV-2.8 GHz with 256 Mbytes RAM.

Table III: Experiment condition.

GOP	IIIII or IPPP
Codec	JM 7.1
MV search range	± 16
QP	10, 16, 24,28,36,40
Number of Reference	1
Common coding option	Hadamard transform, CABAC, RDO is enabled
Size	CIF and QCIF
Number of Frames	300

Comparisons with the case of exhaustive search (RDO) were performed with respect to the change of average PSNR (ΔPSNR), the change of average data bits (ΔBit), and the change of average encoding time (ΔTime), respectively. The PSNR is derived from average PSNRs of luma component (Y) and chroma component (U, V) based on [1].

Also, in order to evaluate the time saving of the proposed fast intra-mode decision algorithm, the following calculation is defined to find the time differences. Let T_{ref} denote the coding time used by JM7.1 encoder and T_{prop} be the time taken by the faster intra-prediction algorithm, and $\Delta\ time$ be defined as:

$$\Delta\, Time\ \ \% \ = \frac{T_{prop} - T_{ref}}{T_{ref}} \times 100 \qquad (6)$$

Also, bitrate increase is defined as:

$$\Delta Bitrate\,\% = \frac{bitrate_{prop} - bitrate_{ref}}{bitrate_{ref}} \times 100 \qquad (7)$$

A group of experiments were carried out on different sequences and the results are shown below. The experiments were ordered in 4 states as listed in Table IV. The encoding bit rates, the PSNR values, and the time saving factor (as compared with the H.264 RDO method) for 4 test sequences with different quantization parameters are shown in Tables V~X. Also, Tables V~VII and VIII~X show the experimental results for IPPPP and IIII sequences, respectively. These tables compare the rate, distortion, and complexity of proposed algorithms with RDO procedure. Generally speaking, as can be seen from this tables, we have saved 60~75% of the total encoding time at the expense of only 1.04% rate increase in average and -0.20dB distortion loss in average for these test sequences. Figures 11 show the examples of RD and the complexity curves of sequences "*Akiyo*" (class A), for IPPP sequences. From this figure, one can see that the proposed fast intra-mode decision scheme gives almost identical RD performance while providing a high speed-up factor (ratio of encoding time using the RDO technique and the proposed scheme). In this figure the RDO, Pan's method, improved Pan's method, and the fast proposed methods are compared (see Table IV). We see that for proposed algorithms the rate-distortion performance loss increases slightly with a high complexity reduction.

Table IV. Different methods of experiment.

category	I4MB	I16MB	Inter-pred. alg.	Early termination
RDO	RDO	RDO	RDO	No
M1	Pan's method	RDO	RDO	Yes
M2	Improved Pan's method	RDO	RDO	Yes
M3	New proposed method	RDO	RDO	Yes
M4	New proposed Method	RDO	New proposed method	Yes

5. CONCLUSION

In this paper using statistical properties of natural video sequences for fast intra-mode decision in the H.264/AVC encoders, we decreased the encoding time by reducing the number of candidate modes. In order to achieve a better performance of computational complexity some new ideas with some strength point of improved Pan's algorithms are combined, and a new algorithm is presented. Also, in this paper, we proposed a fast inter-prediction technique with variable-size bock when utilizing the proposed split and merge procedure. The algorithm is especillay designed for low bit rate storage and transmission. The simulation results show that the proposed algorithm reduces the number of RDO calculations with respect to original and improved algorithms with negligible loss in PSNR and negligible bitrate increase. The proposed algorithms can be used for challenging work of intra- and inter-prediction mode decision in the H.264/AVC video encoders with low computational cost.

Table V. Simulation results for IPPP type sequences. Distortion comparison

		$\Delta PSNR$ (dB)				
		10	16	22	32	40
Foreman	M1	-0.079	-0.069	-0.078	-0.035	-0.050
	M2	-0.510	-0.12	-0.01	0.010	0.100
	M3	-0.12	-0.075	-0.043	-0.150	-0.100
	M4	-0.130	-0.217	-0.01	-0.015	-0.010
News	M1	-0.060	-0.070	-0.24	-0.140	-0.110
	M2	-0.038	-0.010	-0.120	0.003	0.010
	M3	-0.050	-0.052	-0.034	-0.32	-0.110
	M4	-0.21	-0.010	-0.001	-0.010	0.01
Container	M1	-0.089	-0.054	-0.081	-0.076	-0.074
	M2	-0.51	-0.160	-0.1	-0.1	0.00
	M3	-0.080	-0.065	-0.067	-0.040	-0.032
	M4	-0.46	-0.21	-0.01	-0.02	-0.03
Silent	M1	-0.012	-0.025	-0.013	-0.021	-0.025
	M2	-0.042	-0.020	-0.020	-0.019	-0.010
	M3	-0.110	-0.020	-0.010	-0.012	0.001
	M4	-0.030	-0.010	-0.130	-0.010	-0.010

Table VI: Simulation results for IPPP type sequences. Rate comparison

		ΔBit %				
		10	16	22	32	40
Foreman	M1	1.025	1.310	1.212	1.121	1.011
	M2	1.130	1.124	0.810	0.905	0.921
	M3	1.030	1.140	1.110	0.820	1.015
	M4	1.112	0.950	0.904	0.715	0.678
News	M1	1.204	1.121	1.222	1.151	1.024
	M2	1.024	0.740	1.076	0.073	0.095
	M3	0.412	0.732	1.082	0.860	0.637
	M4	1.050	0.042	0.316	0.712	0.810
Container	M1	1.423	1.212	0.490	0.353	0.842
	M2	0.915	1.097	0.912	0.710	0.620
	M3	1.027	0.932	0.910	0.145	0.352
	M4	1.640	0.980	0.732	0.910	0.760
Silent	M1	0.513	0.914	1.072	0.435	0.905
	M2	1.024	0.935	1.045	0.640	0.590
	M3	1.120	0.812	0.970	0.815	0.910
	M4	1.201	0.172	0.625	0.912	0.701

Table VII. Simulation results for IPPP type sequences. Complexity comparison

		Δ *Time* %				
		10	16	22	32	40
Forman	M1	-38.05	-36.42	-31.49	-31.50	-30.45
	M2	-35.23	-37.34	-43.24	-44.25	-45.5
	M3	-42.23	-39.25	-35.24	-35.02	-35.25
	M4	-79.01	-73.02	-74.12	-77.07	-78.32
News	M1	-41.32	-38.24	-35.24	-31.12	-31.10
	M2	-38.45	-39.25	-40.25	-42.13	-43.23
	M3	-43.25	-41.50	-40.21	-39.34	-38.22
	M4	-79.15	-68.27	-67.14	-67.12	-63.40
Container	M1	-31.03	-35.26	-34.02	-33.03	-34.25
	M2	-33.24	-34.25	-37.22	-39.44	-36.25
	M3	-33.24	-34.25	-37.22	-39.44	-36.25
	M4	-61.30	-57.34	-66.34	-64.20	-60.3
Silent	M1	-36.02	-35.01	-31.02	-30.98	-30.37
	M2	-35.46	-37.89	-37.33	-39.42	-41.32
	M3	-40.21	-39.25	-39.10	-38.23	-38.12
	M4	-65.39	-63.79	-63.42	-54.97	-59.32

Table VIII. Simulation results for IIII type sequences. Distortion comparison

		Δ *PSNR*				
		10	16	22	32	40
Forman	M1	-0.060	-0.065	-0.064	-0.063	-0.051
	M2	-0.45	-0.021	-0.01	0.001	0.10
	M3	-0.08	-0.070	-0.062	-0.05	-0.05
	M4	-0.130	-0.122	-0.022	-0.010	-0.01
News	M1	-0.052	-0.060	-0.042	-0.050	-0.051
	M2	-0.037	-0.030	-0.010	0.010	-0.01
	M3	-0.050	-0.051	-0.042	-0.140	-0.010
	M4	-0.022	-0.02	-0.10	-0.02	0.010
Container	M1	-0.059	-0.064	-0.062	-0.056	-0.068
	M2	-0.052	-0.054	-0.201	-0.012	0.001
	M3	-0.182	-0.045	-0.053	-0.048	-0.042
	M4	-0.056	-0.031	-0.042	-0.032	-0.013
Silent	M1	-0.061	-0.054	-0.053	-0.041	-0.035
	M2	-0.050	-0.043	-0.030	-0.023	-0.014
	M3	-0.051	-0.034	-0.042	-0.032	0.020
	M4	-0.043	-0.051	-0.024	-0.064	-0.072

Table IX. Simulation results for IIII type sequences. Rate comparison

		Δ *Bits* %				
		10	16	22	32	40
Forman	M1	1.320	1.340	1.572	1.142	1.370
	M2	1.321	1.234	0.985	0.997	0.980
	M3	1.310	1.410	1.244	0.887	0.547
	M4	1.142	0.993	0.719	0.562	0.539
News	M1	1.132	1.201	1.132	1.215	0.956
	M2	0.970	0.967	0.893	0.742	0.932
	M3	1.130	1.334	0.923	0.950	0.822
	M4	1.120	0.0942	0.876	0.320	0.314
Container	M1	1.203	1.002	1.042	1.095	0.942
	M2	1.310	1.124	1.231	0.981	0.739
	M3	1.312	0.990	0.989	0.720	0.730
	M4	1.345	0.941	0.512	0.710	0.342
Silent	M1	1.223	1.087	0.921	0.905	0.980
	M2	1.750	0.930	1.012	0.590	0.610
	M3	1.940	0.312	0.590	0.435	0.580
	M4	1.920	0.702	0.215	0.736	0.416

Table X. Simulation results for IIII type sequences. Complexity comparison

		Δ *Time* %				
		10	16	22	32	40
Foreman	M1	-47.15	-46.12	-43.29	-49.10	-47.25
	M2	-50.13	-47.10	-52.45.	-54.20	-55.50
	M3	-55.34	-59.25	-45.34	-42.82	-45.15
	M4	-50.05	-51.32	-52.30	-51.27	-54.12
News	M1	-52.24	-58.20	-53.61	-51.90	-50.46
	M2	-49.50	-52.50	-50.15	-52.42	-51.02
	M3	-50.12	-51.23	-50.78	-49.74	-48.89
	M4	-51.34	-52.43	-53.19	-52.10	-52.13
Container	M1	-46.23	-45.32	-44.72	-43.67	-44.39
	M2	-46.42	-48.53	-47.43	-49.64	-46.42
	M3	-45.34	-44.65	-47.39	-49.51	-48.12
	M4	-47.90	-50.20	-51.06	-52.43	-54.10
Silent	M1	-49.21	-46.31	-43.32	-45.38	-43.29
	M2	-49.76	-49.90	-50.03	-49.62	-51.12
	M3	-50.30	-49.50	-50.37	-49.38	-48.92
	M4	-51.22	-58.60	-51.73	-53.38	-54.13

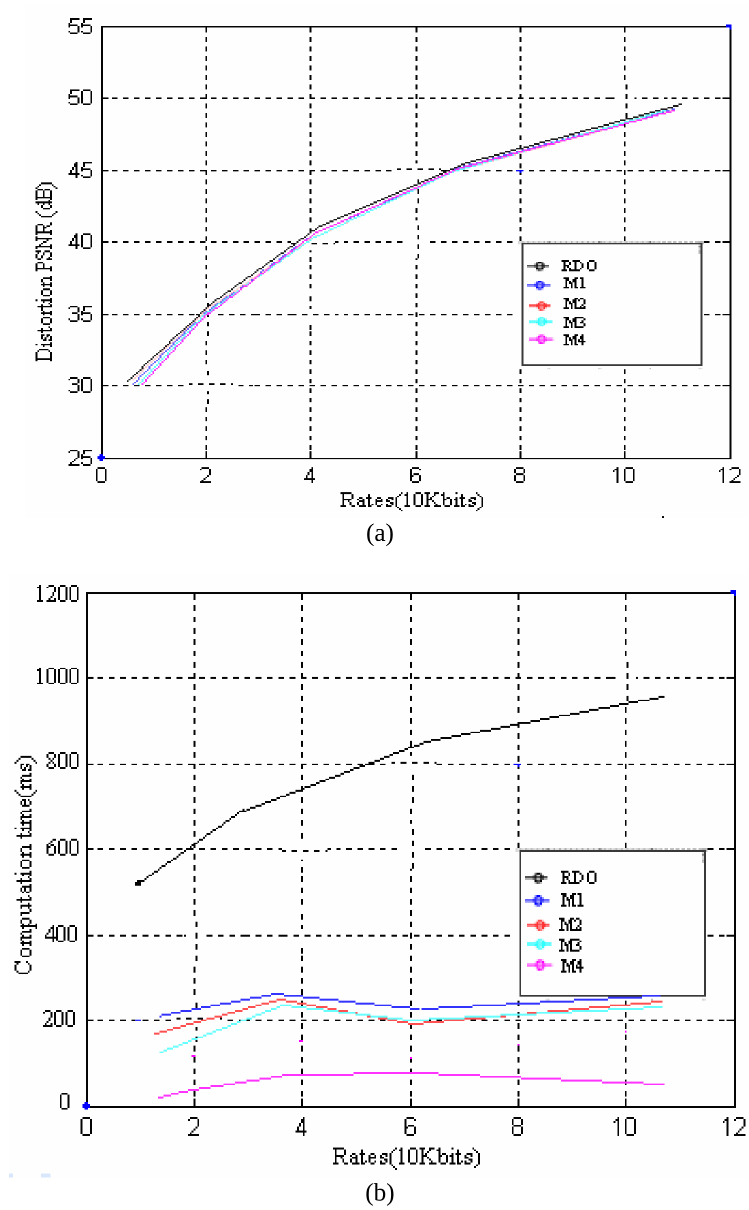


Figure 11: Akiyo sequence, (IPPP seq.). (a) R-D performance, (b) computational complexity.

References

[1] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, and S. Wu, “Fast Mode Decision Algorithm for Intra-prediction in H.264/AVC Video Coding”, IEEE Trans. On circuits and systems for video Tech., Vol. 15, NO. 7, pp. 813-822, July 2005.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC Video Coding Standard”, IEEE Trans. On circuits and systems for video technology, Vol. 13, no. 13, no. 7, pp. 560-576, 2003.

[3] ece.ut.ac.ir/classpages/Multimedia/h264.ppt

[4] Envivio http://www.envivio.com/products/h264.html

[5] Changsung Kim, Qing Li, C. C. Jay Kuo, “Fast Intra-Prediction Model Selection for H.264 Codec,” SPIE International Symposium ITCOM 2003, Orlando, Florida, Sept. 7-11, 2003.

[6] F. Pan, X. Lin, *et al*., “Fast Mode Decision for Intra- Prediction,” ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT 7*th* Meeting, Pattaya II, Thailand, March 2003.

[7] F. Pan, X. Lin, S. Rahardja, K. P. Lim, and Z. G. Li, “A Directional Field Based Fast Intra-Mode Decision Algorithm for H.264 Video Coding,” *IEEE Inter. Conf. on Multimedia and Expo*, vol. 2, pp. 1147-1150, June 2004.

[8] J. Kim and J. Jeong, “Fast Intra-Mode Decision in H.264 Video Coding Using Simple Directional Masks”, Proc. of SPIE Vol. 5960, pp 1071-1079, 2005.

[9] R. Garg, M. Jindal, M. chauhan, “Statistics Based Fast Intra-Mode Detection”, Proc. of SPIE Vol. 5960, pp 2085-2091, 2005.

[10] B. Jeon and J.lee, “Fast Mode Decision for H.264”, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT 10th Meeting, Waikoloa, Hawaii, December 2003.

[11] C. Yang, L. PO, W. Lam, “ A Fast H.264 Intra Prediction Algorithm Using Macroblock Properties”, ICIP, pp. 461-464, 2004.

[12] B. Meng, O. C. AU, C. Wong, H. Lam, “ Efficient Intra-Prediction Algorithm in H.264”, IEEE, pp. 837-840, 2003.

[13] C. Tseng, H. Wang, J. Yang, “ Improved and Fast Algorithms for Intra 4x4 Mode Decision in H.264/AVC“, IEEE, pp. 2128-2131, 2004.

[14] C. Hsu, M. Ho, J. Hong, “ An Efficient Algorithm for Intra-prediction in H.264”, IEEE, pp. 35-36, 2006.

[15] Joint Video Team (JVT), reference software JM7.1, http://bs//hhi.de/~suehring/tml/download/JM7.1.zip.

[16] M. Jafari, S. Kasaei, “ An Efficient Intra Prediction Mode Decision Algorithm for H.263 to H.264 Transcoding”, IEEE international conference on computer systems and applications, page 1082-1089, march 2006.

[17] M. Jafari, s. Kasaei, “ Prioritisation of data partitioned MPEG-4 video over GPRS/EGPRS mobile networks”, Asian internet engineering conference (AINTEC), Tailand, pp. 68-82, December 2005.

[18] I. Richardson, “ H.264/MPEG-4 part 10 white paper”, available: http://www.vcodex.fsnet.co.uk/resources.html.

[19] Jiang Gang-yi, Li Shi-ping, Yu Mei, Li Fu-cui, “AN EFFICIENT FAST MODE SELECTION FOR INTRA PREDICTION”, IEEE Int. Workshop VLSI Design & video Tech., China, pp. 357-360, May 28-30, 2005.

[20] Yu-wen Huang, Bing-Yu Hsieh, Shao-Yi Chien, Shyh-yih Ma, and Liang-Gee Chen, “Analysis complexity reduction of multiple reference frames motion estimation in H.264/AVC”, February 2006.

[21] Yin, P., Tourapis, H.C., Tourapis, A.M., Boyce, J., “Fast mode decision and motion estimation for JVT/H.264. proc. IEEE Int. Conf. Image processing”, Barcelona, Spain. 3 (2003) 14-17.

[22] Lim, K.P., Wu, S., Wu, D.J., Rahardja, S.,Lin, X., Pan, F., Li, Z.G.,” Fast inter mode selection. Joint video team of ISO/IEC MPEG and ITU-T VCEG, JVT-1020 (2003).

[23] Andy Chang; Au, O.C.; Yeung, Y.M.; .,” A novel approach tofast multi-block motion estimation for H.264 video coding“, international conference on multimedia and expo, July 2003.

[24] G. Sullivan, “Recommended simulation common conditions for H.26L coding efficiency experiments on low resolution progressive scan source material,” presented at the 14th VCEG-N81 meeting, Santa Barbara, CA, Sep. 2001.