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

Mehdi Jafari† **and Shohreh Kasaei**††,

Islamic Azad University, Kerman Branch, Sharif university of Technology

Summary
H.264/AVC, the latest video standard, adopts *rate-distortion optimization* (RDO) technique to obtain the best intra- and inter-prediction, while maximizing visual quality and minimizing the required bitrate. However, the full RD cost calculation for all intra-prediction modes, the exhaustive searches for finding optimal motion vectors for all block sizes, and the multiple references frame procedure considerably increase its computational complexity with the allowed number of prediction modes. In order to reduce the complexity, here we propose a new approach for both inter- and intra-mode decisions, that takes into account the two effective parameters, image content type and the quantization parameter. The proposed fast intra-prediction mode selection strategy uses some observations on the interior and the exterior MB properties to select a subset of candidate modes at different quantization parameters. Also, the fast inter-prediction mode decision approach uses split/merge procedure based on correlation of motion vectors and motion details of video objects. Also, here we use a context-based adaptive method to speed up the multiple reference frames motion estimation that is based on inter- and intra-prediction residues and quantization parameters. As such, only a subset of inter- and intra-modes is chosen for RDO calculation. Experimental results show that the proposed algorithm, reduces the total encoding cost with negligible loss in PSNR and a slightly increase in the required bitrate, when compared to RDO and other fast algorithms reported in the literature.

Key words:
H.264/AVC, intra- and inter-prediction, rate-distortion optimization, similar predicted-pixels, split/merge.

1. Introduction

As recent multimedia applications 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, minor increase in decoder complexity compared to existing standards, adaptation to delay constraints, error robustness, and network friendliness [1, 2]. Table 1 [3] and Figure 1 [4] show some performance comparisons using MPEG-2, MPEG-4 (ASP), and H.264/AVC. To achieve an outstanding coding performance, H.264/AVC employs several powerful coding techniques such as directional prediction of intra-coded blocks, inter-prediction with variable block-size motion compensation, multi-reference frame motion estimation, motion vectors with quarter-pel accuracy, in-loop deblocking filter, 4×4 integer transform, and the forth. According to these new features, the encoder computational complexity has extremely increased compared to the previous standards. This makes H.264/AVC difficult for applications with low computational capabilities. Thus, until now, the reduction of its complexity is a challenging task.

Table 1: Average bit-rate reduction compared to prior coding schemes[3].

Standards	MPRG-4/ASP	H.263/HLP	MPEG-2
H.264/AVC	38.62%	48.80%	64.46%
MPRG-4/ASP	---	16.65%	42.95%
H.263/HLP	---	---	30.61%

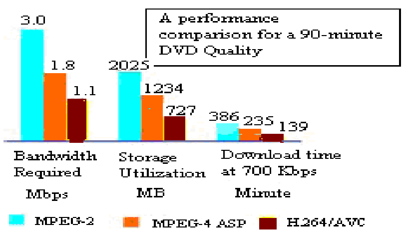


Fig. 1 Performance comparison of different video coding standards [4].

Like previous standards, H.264/AVC still uses motion compensated transform coding. The improvement in coding performance comes mainly from the intra- and inter-prediction part. In particular, intra-prediction at different block sizes and an improved number of directional mode decisions contribute to a better video quality increase. Also, inter-prediction using *motion estimation* (ME) at quarter-pixel accuracy with variable block sizes and multiple reference frames greatly reduces the prediction error. H.264/AVC employs the Lagrange RDO method to find out the best coding mode of intra-and inter-prediction with highest coding efficiency. The RDO technique requires a lot of computations since it tests the encoding process with all possible coding modes of intra- and inter-coding, and calculates their RD costs to choose

Manuscript received May 5, 2008
Manuscript revised May 20, 2008

the mode having the minimum required bitrate. 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 2 [20]. As shown in Figure 2, motion estimation is the most computationally intensive part. The instruction profile of the reference software shows that real-time encoding of CIF 30Hz video (baseline options, search range [-16.75, +16.75], five reference frames) requires 314,994 million instructions per second and memory access of 471,299 MBytes/sec.

Also, H.264/AVC offers a rich set of prediction patterns for intra-prediction, *i.e.*, nine prediction modes for 4×4 luma blocks and four prediction modes for 16×16 luma blocks. 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]. Thus, the computational burden of these types of brute force-searching algorithm is far more demanding than any existing video coding algorithm.

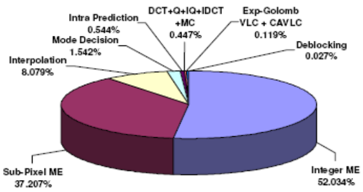


Fig. 2 Run-time percentages of functional blocks in H.264/AVC baseline encoder [20].

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 have been proposed. For fast inter-prediction mode decision, the early termination technique [23] reduces the number of potential prediction modes. In [24], a classification method is proposed. Recently fast inter-mode selection algorithms were proposed in [25], [26] and [27] to alleviate the encoder complexity due to the motion estimation and inter-mode decisions. In [28], a fast inter-prediction mode decision based on pre-encoding process is presented.

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. Also, 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 bitrate 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.

This paper describes a new approach for both inter-and intra-mode size selection, that takes into account the content of the video and the quantization parameter used to encode each macroblock. In fact, the mode selection also depends on the quantization parameter, since the video quality due to different quantization levels influences the choices on different encoding tools.

Here, for intra-prediction mode decision, we improve Pan's method [1, 6, 7]. As an alternative method, here we proposed a fast intra-method using statistical properties of adjacent MBs and reference pixels with a combination of the presented algorithms in Pan's method.

Also, for inter-prediction mode decision, the split/merge procedure is used. In this method a macroblock is divided into equal-size quarters, and then using the similarities of motion vectors of adjacent blocks we show how to merge the sub-blocks for quarter divisions.

In this paper, an effective method for accelerating the multiple reference frames ME without significant loss of video quality is proposed, which is based on analyzing the available information obtained from previously processed frames. We have verified the different parts of the proposed algorithm step by step by implementing it on JM7.1 [15] reference software and have compared its performance with RDO and other available fast algorithms. Experimental results show that the proposed method reduces the encoding cost up to 40% with a negligible loss in the reconstructed video quality and a negligible increase in the required bitrate.

The remaining parts of the paper are organized as follows. We review the full search RDO intra- and inter-prediction scheme of H.264/AVC in Section 2. Section 3 and 4 present the fast intra- and inter-prediction mode decision algorithms, respectively. Also, complexity reduction of multiple reference frames motion estimation is considered in Section 4. Experimental results are given in Section 5 and finally Section 6 concludes the paper.

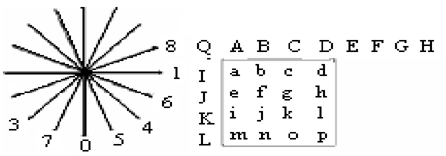
2. Mode Decision and RDO in H.264/AVC

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.

2.1 Intra-Prediction Mode Decision

Intra-prediction is based on the observation that adjacent macroblocks tend to have similar properties. Prediction may be formed for each 4×4 luma block (I4MB), 16×16 luma MB (I16MB), and 8×8 chroma block.

For prediction of 4×4 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 3(a). In Figure 3(b), the block (values of pixels “a” to “p”) is to be predicted using A to Q pixel values.



(a) (b)

Fig. 3 (a) Intra-prediction modes for 4×4 luminance blocks. (b) Labeling of prediction samples.

The DC prediction (mode 2), useful for those blocks with little or no local activities, the other modes (1-8) may only be used if all of the required prediction samples are available.

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

H.264/AVC supports four chroma prediction modes for 8×8 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).

2.2 Inter-Prediction Mode Decision

Inter-prediction is based on using motion estimation and compensation to take advantage of the temporal redundancies that exist between successive frames. The important differences from earlier standards include the support for a range of block sizes (down to 4×4), multiple reference frames, and fine sub-pixel motion vectors (1/4 pixel in the luma component).

H.264/AVC supports motion compensation block sizes ranging from 16×16 to 4×4 luminance samples with many options between the two. The luminance component of each macroblock (16×16 samples) may be split up in 4 ways as 16×16, 16×8, 8×16 or 8×8. If the 8×8 mode is chosen, each of the four 8×8 macroblock partitions within the macroblock may be split in a further 4 ways as 8×8, 8×4, 4×8 or 4×4. These partitions and sub-partitions give rise to a large number of possible combinations within each macroblock.

A separate motion vector is required for each partition or sub-partition. Each motion vector must be coded and transmitted; in addition, the choice of partition(s) must be encoded in the compressed bitstream. Choosing a large partition size (*e.g.,* 16×16, 16×8, 8×16) means that a small number of bits are required to indicate the choice of motion vector(s) and the type of partition; however, the motion compensated residual may contain a significant amount of energy in frame areas with high details. Choosing a small partition size (*e.g.,* 8×4, 4×4, etc.) may give a lower energy residual after motion compensation, but requires a larger number of bits to signal the motion vectors and the choice of partition(s). The choice of partition size therefore has a significant impact on compression (a small partition size may be beneficial for detailed areas).

H.264/AVC as enhanced reference picture selection as H.263++ enables efficient coding by allowing an encoder to select, for motion compensation purposes, among a large number of pictures that have been decoded and stored in the decoder.

2.3. RDO Procedure

For I-frames 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, intra- and inter-prediction is done and RDO is used to find the best prediction. The RDO procedure to encode one MB in an I-frame is given in [5].

According to the RDO procedure of intra-prediction in H.264/AVC, the number of mode combinations for luma and chroma blocks in a macroblock is $N8\times(16\times N4 + N16)$, where *N8*, *N4*, and *N16*, denote the number of modes for 8×8 chroma blocks, and 4×4 and 16×16 luma blocks, respectively [5]. Also, according to the RDO procedure of inter-prediction, assume that we have *M* block modes, *N* reference frames, and $\pm W$ search range. Then, we need to check $M.N.(2W+1)^2$ positions for a single reference frame and single block mode. This makes the complexity of the encoder extremely high. In order to reduce the encoding complexity with little RD performance degradation, fast intra- and intra-prediction mode decision methods are proposed.

3. Proposed Fast Intra-Prediction Mode Decision Methods

This section presents a new fast intra-prediction algorithm. The proposed method is based on several facts that we have observed from the statistics of different sequences as follows:

a) Figure 4 shows the total number of 4×4 and 16×16 intra-coded macroblocks at different *quantization parameters* (QPs). As a result, fast detection of 4×4 intra-prediction mode can significantly improve the encoding speed at low QPs, while 16×16 intra-prediction can improve the speed at large QPs.

b) The prediction modes of each block are correlated with those of neighboring 4×4 luminance blocks.

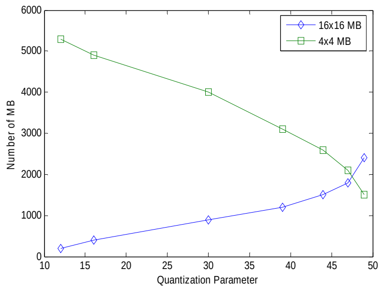


Fig. 4 Number of 4×4 and 16×16 intra-coded macroblocks at different quantization parameters.

c) Normally, pixels along the direction of local edges have similar values. Therefore, a good prediction can be achieved if we predict the pixels using those neighboring pixels that lie in the same directions of the edges.

d) The optimal mode (found by full-search) and other "good" (second or third best) modes are most likely to have similar directions.

e) The directional features of 4×4 blocks can be preserved roughly after down-sampling.

f) Experimental results show that the reference pixels of a 4×4 luma block are likely to be similar to each other [22].

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 Improved Pan's Method for Fast Decision of I4MB

Pan et al. presented a fast mode selection for intra-prediction method in [1] in which the average edge direction of a given block is measured. The Sobel operators are first used to obtain the directional vector of each pixel in a block by:

$$\vec{D}_{i,j} = \{dx_{i,j}, dy_{i,j}\} \tag{1}$$

where the Sobel operator are:

$$dx_{i,j} = P_{i-1,j+1} + 2\times P_{i,j+1} + P_{i+1,j+1} - P_{i-1,j-1} - 2\times P_{i,j-1} - P_{i+1,j-1}$$
$$dy_{i,j} = P_{i+1,j-1} + 2\times P_{i+1,j} + P_{i+1,j+1} - P_{i-1,j-1} - 2\times P_{i-1,j} - P_{i-1,j+1} \tag{2}$$

The amplitude and angle of each edge vector can be calculated using

$$Amp\ (\vec{D}_{i,j}) = |dx_{i,j}| + |dy_{i,j}| \tag{3}$$

and,

$$Ang\ (\vec{D}_{i,j}) = \frac{180^{o}}{\pi} \times \arctan(\frac{dy_{i,j}}{dx_{i,j}}) \tag{4}$$

When *Ang*(.) is fitted into one of the 8 modes.

Then, the edge directional histogram of the block is found . The *edge direction histogram* (EDH) indicates the number of pixels with similar edge directions. Therefore, the cell *k* with the maximum amount indicates that there is a strong edge along that direction in the block and thus assigned as the dominant block direction. Figure 5 show the edge direction histogram of Figure 6 and an example of edge dominant direction, respectively.

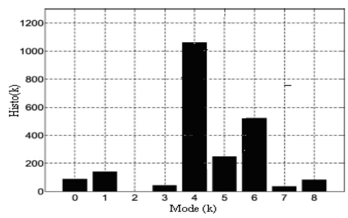


Fig. 5 Edge direction histogram of Figure 5.

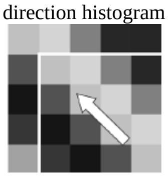


Fig. 6 An example of 4×4 edge patterns and their dominant direction.

In the 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 16×16 luma block and 8×8 chroma block. Here, to improve the Pan's method, we eliminate the DC mode from the candidates if the direction of the block is obvious, and otherwise, we choose only DC mode. To check whether the DC of the block is clear or not, the *diff* value, given in Eq. (5), is obtained to check whether it is smaller than a predetermined threshold or not, using

$$diff = \sum_{i=0}^{i=15} | avg - p_i | \qquad (5)$$
$$avg = (\sum_{i=0}^{15} p_i + 8) >> 4$$

The improved Pan's method is proposed as follows:

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

$$diff1 = \sum_{i=0}^{i=64} | avg - p_i | \qquad (6)$$
$$avg = (\sum_{i=0}^{64} p_i + 32) >> 6$$

5. If *diff1* < T1, choose the maximum prediction mode and the DC mode. Extract the maximum prediction mode as I4MB but with DC and only 3 directions of intra16.

6. For 8×8 chroma block, and after down-sampling by a factor of 2, use the same procedure as I16MB but by using Eq. (13).

Pan's method can reduce RDO calculation from 592 to 132 times. The number of candidate modes and the RDO calculation in the worst and the best cases are shown in Table 2.

Table 2. Number of candidate modes

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

Table 2 summarizes the number of candidates selected for RDO calculation based on edge direction histogram. As can be seen from this table, 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).

3.2 Fast Intra-Prediction Mode Selection based on Statistical Properties of Adjacent MBs and Reference Pixels

The method proposed in this section analyzes the characteristic of reference pixels, and uses the similarity between adjacent MBs, while the improved Pan's method analyzes the characteristic of the 4×4 block itself. As a result, the combination of these three kinds of methods can achieve better results. The proposed algorithm is as follow:

1. Find the maximum value of the edge directional histogram H. Denote the corresponding mode by M1.
2. If the modes for one of the top or left blocks are M1, then choose M1 as the best candidate mode for the current block. Go to step 7.
3. For 4×4 luma block, compute the *mean of absolute difference* (MAD) of its reference pixels. Then, if this value is smaller than a predetermined threshold, select M1. Go to step 7.

This result is yielded from the 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 the *mean of absolute difference of horizontal references* (MADH) is less than a threshold and M1 is a member of the set {mode 0, mode 3, mode 7}, select M1 is. Go to step 7.
5. Also, if the *mean of absolute difference of vertical references* (MADV) is less than a threshold and M1 is a member the set of {mode 1, mode 8}, select M1. 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, use the improved Pan's method (explained in Section 3.1).
7. Terminate.

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

To increase the speed of the algorithm, we use the early termination of RDO calculations for all proposed algorithms as in [1].

4. Proposed Fast Inter-Prediction Mode Decision Method

The motivated facts about the inter-prediction are summarized below.

a) The block with high motion activities, instead of high textural details, can be better coded using smaller block sizes, while the block with less motion activities 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.Then homogenous regions are encoded using 16×16 block sizes while non-homogenous regions are encoded using smaller block sizes.
c) The background is not homogeneous, but because of the temporal stationary is coded using 16×16 block size.

4.1 Fast Inter-Prediction Mode Decision using Split/Merge Procedure

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

1. Subtract the current frame from its previous frame, then for any 16×16 MB compute:

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

2. If N_{im}/N_i is smaller than or equal to a predetermined threshold, choose direct mode as the final macroblock type.
3. Otherwise, split the block into four 8×8 blocks and conduct a new iteration of block matching for each of these four descending blocks.
4. If motion vectors of 8×8 sub-block are equal or three sub-block MV are the same and the forth unequal MV only differ by one quarter-pixel distance, choose mode 1(16×16). Terminate.
5. If MV0=MV1 and MV2=MV3 (see Figure 7), choose the 8×16. Terminate.
6. If MV0=MV2 and MV1=MV3, choose the 16×8. Terminate.
7. Repeat the steps 2 to 4 for each 8×8 blocks, except that the sub-blocks are 4×4.
8. Terminate.

MV0	**MV1**
MV2	**MV3**

Fig. 7 MB division.

The mode selection methodology employed in this paper is as below:
For I4MB, I16MB and P the proposed algorithms are used to extract the best mode among the related category and at last RDO is used to extract the final mode.

4.2 Analysis and Complexity Reduction of Multiple Reference Frames Motion Estimation

In H.264/AVC, motion estimation is allowed to search multiple reference frames. Therefore, the required computation is highly increased, and it is in proportion to the number of searched reference frames.
Experimental results show some facts that are used for deciding the number of references frames as[20]:
1. For *QP*=20, *QP*=30 and 40, it can be seen that 65%, 79% and 95%, of macroblocks need only one reference frame, respectively. Therefore, we should proceed the block matching process from the nearest reference frame to the farthest reference frame.
2. Another interesting point is that low bitrate cases are more likely to have best reference frames close to the current frame than higher bitrate cases are.
3. We can see that for *QP*=20, there are 59.84%, 05.00%, 04.88%, 28.11%, and 02.17% of the macroblocks selected as P16×16, P16×8, P8×16, P8×8, and intra, respectively, when only one previous frame is searched. For *QP*=30, there are 75.97%, 05.36%, 05.45%, 11.04%, and 02.18% of the macroblocks selected as P16×16, P16×8, P8×16, P8×8, and intra, respectively. For *QP*=40, the corresponding percentages are 89.34%, 03.21%, 03.07%, 01.69%, 02.69%.
4. In H.264/AVC, SKIP mode is a special case of P16×16. The percentages of SKIP macroblocks after one reference frame is searched are 44.57%, 62.69%, and 79.14% for *QP*=20, 30, and 40, respectively.
This result show, a large percent of MBs are coded as 16×16 or skip mode, that are use only one reference frame, while for large QP this fact are amplified. According to these observations, in the following, we list the steps for each macroblock to check whether it is necessary to search the next reference frame at the end of each reference frame loop.
1. After the prediction procedure, residues are transformed, quantized, and then entropy coded. If we can detect the situation that the transformed and quantized coefficients are very close to zero in the first reference frame, we can turn off the matching process for the remaining frames.
2. Calculate the sum of absolute transform difference (SATD), if it is less than threshold(*THSATD)*, we will stop the searching process.
3. If the best reference frame is previous frame and if the best motion vector is the same as that of SKIP mode or

Category	Intra- prediction (I4MB,I16MB Chroma)	Inter-Prediction	Multi-Reference Algorithm	Early Termination
RDO	RDO	RDO	RDO	YES
M1	Pan's Method	RDO	RDO	YES
M2	Improved Pan's method	RDO	RDO	YES
M3	Proposed Alg.	RDO	RDO	YES
M4	Proposed Alg.	Proposed Alg.(split/ merge)	RDO	YES
M5	Proposed Alg.	Proposed Alg.(split/ merge)	Proposed Alg.	YES

16×16 mode and QP is larger than threshod(*THQP)*, the multiple reference frames loop will be early terminated. The determination of *THQP* is empirically obtained in [20]. In the proposed method 76%-96%of computation for searching unnecessary reference frames can be avoided.

Also, similar to intra-prediction we use an early termination technique based on early detection of zero blocks.

5. Experimental Results

Our proposed algorithm was implemented into JM7.1, provided by JVT according to the test conditions specified in VCEG-N81 document as listed in Table 3 [21]. Experiments were carried out on the recommended sequences with various quantization parameters for IPPP… 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. We compared the performance of our proposed algorithm (fast motion estimation + fast intra + fast inter + selective frame) with other available approaches. The used fast motion estimation algorithm is from JVT-F017, the UMHexagonS algorithm (combined with the *center biased fractional-pel search* (CBFPS) and early termination technique) was integrated within version 7.1 of the JVT software [15]. To show the impact of different parts of the algorithm, these parts are added at different steps and the results are analyzed. Thus, the experiments were ordered in six states as listed in Table 4.

Table 3: Experiment conditions.

GOP	IIIII or IPPP
Codec	JM 7.1
MV Search Range	±16
Quantization Parameter	10, 16, 24,28,36,40
Number of References	5
Common Coding Option	Hadamard Transform, CABAC, RDO is enabled
Format	CIF and QCIF
Number of Frames	300

Table 4. Different methods used in the experiments

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

$$\overline{PSNR} = 10 \log_{10}\left(255^2 / MSE\right) \tag{7}$$

Therefore, in the rest of this paper we use the overall PSNR value of all three components Y, U and V using Equ. (11).

$\Delta\ time$ be defined as:

$$\Delta Time\ \% = \frac{T_{prop} - T_{ref}}{T_{ref}} \times 100 \quad . \tag{8}$$

Also, the bitrate increase is defined as:

$$\Delta Bitrate\ \% = \frac{bitrate_{prop} - bitrate_{ref}}{bitrate_{ref}} \times 100 \quad . \tag{9}$$

A group of experiments were carried out on different sequences and the results are shown below. The encoding bitrates, 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 5~7. Generally speaking, as can be seen from this tables, we have saved 40-50% of the total encoding time at the expense of only 0.1~1.5% rate increase in average and 0.015 distortion in average for these test sequences.

Figures 8, 9, and 10 show the examples of RD and the complexity curves of sequences "*Akiyo*" (class A), "*Foreman*" (Class B), and "*Stefan*" (Class C) for IPPP sequences. From these figures, one can see that the proposed fast decision scheme gives almost identical rate-distortion performance while providing a speed-up factor (ratio of encoding time using the RDO technique and the proposed scheme) of 3-6. In this figure, the RDO, Pan's method, improved Pan's method, and 3 forms of fast proposed methods (Fast intra + Fast inter + Fast multi reference frames) are compared (see Table 4). Figures 11, 12, and 13 give examples of the classification results and the final mode decision for "*Football*", "*Stefan*" and "*walking person*"' sequences. In these figures, the red, yellow, and green colors show the skip, intra- mode, and inter-mode, respectively. In part (c) of these figures, the black blocks are the macroblocks with inter-prediction and part (d) shows the motion vector and intra-prediction mode for each block.

6. Conclusion

In this paper, an efficient intra- and inter-mode decision algorithm for H.264/AVC standard was proposed. In the proposed algorithm, we decreased the encoding time by reducing the number of candidate modes. In order to achieve a better performance, some new ideas (such as similarity of references pixel, adjacent MB properties, and statistical properties of variable block sizes) with some strength points to improve Pan's algorithm were combined. Also, a fast ME algorithm was used to reduce the encoding cost.

In order to evaluate the impact of different parts of the algorithm, they were added step by step and the related experimental results were shown until the final algorithm was constructed. The experimental results show that the proposed algorithm has reduced the number of RDO calculations with respect to the original and improved algorithms with negligible loss in reconstructed video quality and a negligible bitrate increase. As the experimental results show, the proposed algorithm can be used for challenging work of intra-prediction mode decision in the H.264/AVC video encoders with low computational cost.

Acknowlegment

This work was in part supported by a grant from ITRC.

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-4 $\times$ 4 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", 10th 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] C. Kim, H.-H. Shih, and C.-C. J. Kuo, "Feature-based intra-prediction mode decision for h.264," in *Proc. IEEE International Conference on Image Processing*, 2004.

[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] 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.

[22] 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.

[23] Yin, P., Tourapis, H.C., toupis, A.M., and boyce, J.: 'Fast mode decision and motion estimation for JVT/H.264'. Proc. Int. Conf. on Image Processing, Barcelona, Spain, 2003, Vol.3, pp. 14-17.

[24] Lim, K.P., Wu, S.,Wu, D.j.Rahardja, S., Lin, X., Pan, f., and Li, Z.G.:' Fast inter mode selection'. 9th Joint video Team(JVT) Mtg, San Diego, CA, USA, September 2003.

[25] H. Kim, Y. Altunbasak, "Fast Mode Decision for Inter Prediction in H.264," ICIP 2004, Singapore, 2004.

[26] Y.H. Kim, J.W. Yoo, S.W. Lee, J. Shin, "Adaptive mode decision for H.264 encoder," in Electronic Letters, Vol.40 No.19, September, 2004.

[27] A. C. Yu, G. Martin, "Advanced block size selection algorithm for inter frame coding in H.264/MPEG-4 AVC," in ICIP 2004, Singapore, 2004.

[28] Q. Dai, D. Zhu, R. Ding, "Fast mode decision for inter prediction in H.264," in ICIP 2004, Singapore, 2004.

Table 5. Experimental results for IPPP type sequences, distortion comparison

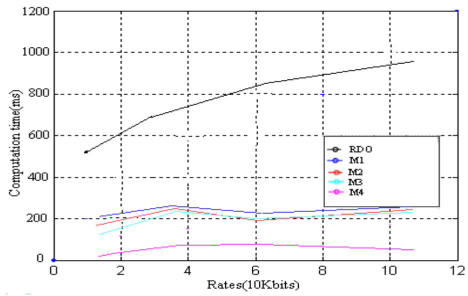
		Δ *PSNR* (*dB*)				
		10	**16**	**22**	**32**	**40**
Foreman	**M1**	-0.081	-0.079	-0.077	-0.065	-0.061
	M2	-0.5	-0.2	-0.1	0.0	0.1
	M3	-0.09	-0.081	-0.073	-0.05	-0.05
	M4	-0.30	-0.27	-0.012	-0.015	-0.01
	M5	-0.012	-0.01	-0.012	-.002	-.001
News	**M1**	-0.073	-0.071	-0.067	-0.064	-0.062
	M2	-0.047	-0.023	-0.01	0.00	0.0
	M3	-0.061	-0.060	-0.059	-0.50	-0.001
	M4	-0.03	-0.01	-0.00	-0.01	0.01
	M5	-0.020	-0.13	-0.016	-0.013	-0.120
Container	**M1**	-0.089	-0.083	-0.081	-0.076	-0.074
	M2	-0.51	-0.17	-0.1	-0.1	0.00
	M3	-0.080	-0.065	-0.067	-0.069	-0.032
	M4	-0.46	-0.21	-0.01	-0.02	-0.03
	M5	-0.10	-0.204	-0.013	-0.013	-0.032
Silent	**M1**	-0.032	-0.035	-0.033	-0.032	-0.029
	M2	-0.04	-0.037	-0.023	-0.019	-0.01
	M3	-0.1	-0.02	-0.01	-0.02	0.00
	M4	-0.03	-0.01	-0.23	-0.012	-0.013
	M5	-0.324	-0.014	-0.080	-0.001	-0.035

Table 6: Experimental results for IPPP type sequences, rate comparison.

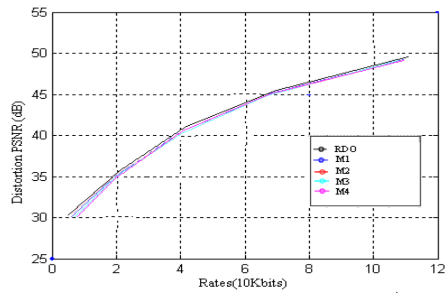
		Δ *Bit* %				
		10	**16**	**22**	**32**	**40**
Foreman	**M1**	1.650	1.540	1.536	1.354	1.230
	M2	1.050	1.004	0.962	0.987	0.870
	M3	1.230	1.210	1.035	0.670	0.345
	M4	1.032	0.890	0.634	0.425	0.478
	M5	1.130	0.735	0.917	0.890	0.098
News	**M1**	1.534	1.001	1.022	1.030	0.924
	M2	0.924	0.940	0.876	0.830	0.910
	M3	1.021	0.932	0.982	0.760	0.567
	M4	1.120	0.0942	0.876	0.320	0.314
	M5	1.098	0.954	0.897	0.830	0.210
Container	**M1**	1.803	1.902	1.090	0.950	0.921
	M2	0.983	0.987	0.732	0.510	0.321
	M3	1.673	0.982	0.879	0.340	0.450
	M4	1.345	0.941	0.512	0.710	0.342
	M5	2.340	0.980	0.987	0.604	0.324
Silent	**M1**	0.923	0.987	0.875	0.875	0.745
	M2	1.624	0.720	0.945	0.742	0.439
	M3	2.100	0.789	0.870	0.615	0.370
	M4	2.301	0.872	0.425	0.576	0.346
	M5	1.250	0.99	0.612	0.512	0.367

Table 7. Experimental results for IPPP type sequences, complexity comparison.

		Δ *Time* %				
		10	16	22	32	40
Foreman	M1	-37.05	-35.42	-33.49	-32.60	-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	-39.95	-41.32	-41.23	-43.47	-44.32
	M5	-43.50	-42.23	-42.12	-40.25	-38.25
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	-41.25	-42.02	-43.02	-43.24	-44.23
	M5	-40.24	-40.34	-39.49	-39.56	-38.97
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	-37.88	-39.50	-41.56	-43.23	-44.41
	M5	-41.25	-40.45	-40.37	-39.33	-38.01
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	-37.30	-38.50	-39.23	-41.78	-42.73
	M5	-41.34	-39.55	-39.33	-39.10	-38.28

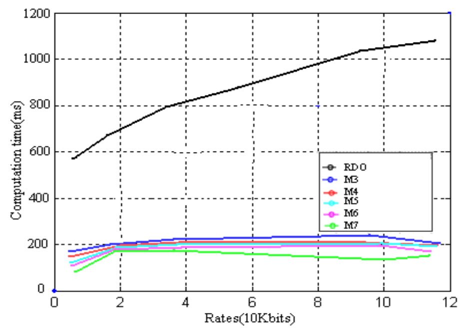


(a)

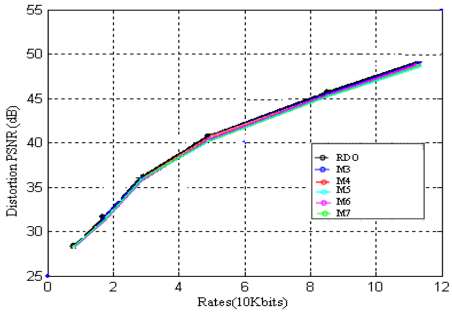


(b)

Fig. 8 "*Akiyo*" sequence (IPPP seq.). (a) Computational complexity. (b) R-D performance.

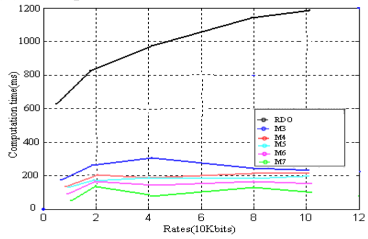


(a)

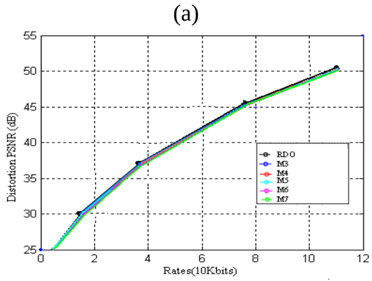


(b)

Fig. 9 "*Foreman*" sequence (IPPP seq.). (a) Computational complexity. (b) R-D performance

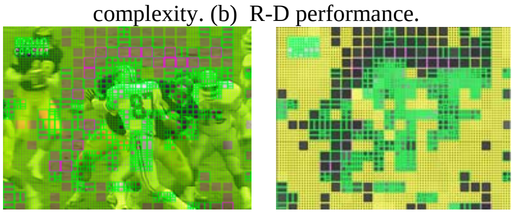


(a)

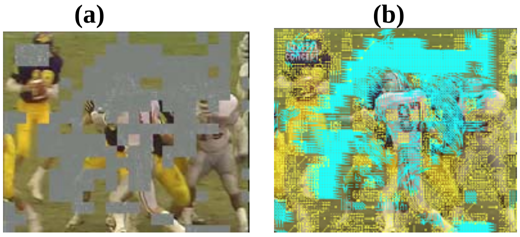


(b)

Fig. 10 "*Foreman*" sequence (IPPP seq.). (a) Computational complexity. (b) R-D performance.

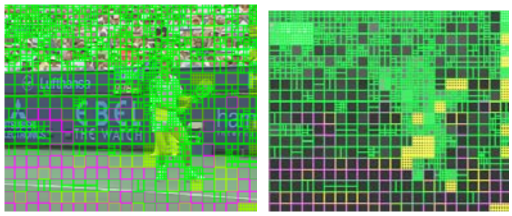


(a) **(b)**

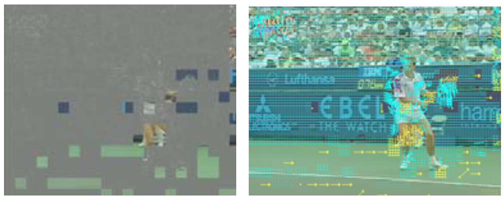


© **(d)**

Fig. 11 Samples of "*Football*" sequence. (a) MB division. (b) MB division, without background. (c) Classification result of intra- and inter-prediction. (d) Motion vector, and intra-prediction modes.

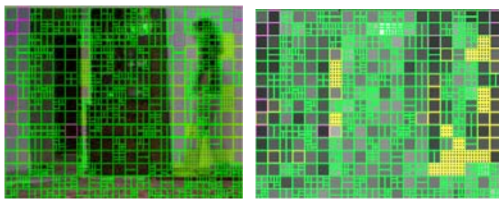


(a) **(b)**

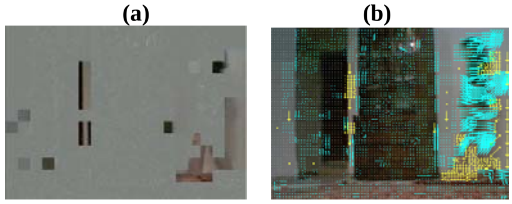


© **(d)**

Fig. 12 Samples of "*Stefan*" sequence. (a) MB division. (b) MB division, without background. (c) Classification result of intra- and inter-prediction. (d) Motion vector, and intra-prediction modes.



(a) **(b)**



(c) **(d)**

Fig. 13 Samples of "*walking person*". (a) MB division. (b) MB division, without background. (c) Classification result of intra- and inter-prediction. (d) Motion vector, and intra-prediction modes.

M. Jafari received the B.S. in electronic engineering and M.S. degrees in communication systems from Shiraz university of technology in 1992 and 1996, respectively. During 1992-1998 he stayed in signal processing research laboratory, ministry of telecommunications of Iran. He received Ph. D degree in communication systems in science and research branch of Islamic Azad University in January 2008. He is currently head of electrical engineering department of Islamic Azad university, Kerman Branch. His main research interests are video compression, image and video processing, pattern recognition and prediction.

Shohreh Kasaei received B.S. in electronic engineering (1977-1986), M.S. degrees in image processing(1992-1994), Ph.D in image processing(1996-1998), in Isfahan university of technology, university of Ryukyus, Okinawa, Japan and Queensland university of technology Australia. During 1999-present, she is as associate professor of Sharif university of technology. Her main research interests are Image and video processing with primary emphasis on object–based video compression, video surveillance, content–based image retrieval, video restoration, motion estimation, fingerprint authentication/ identification, virtual studios, object tracking, color image processing, hyperspectral change detection, multidimensional signal modeling and prediction. Also, multiscale analysis with application to video compression, image enhancement, pattern recognition, motion tracking, texture segmentation and classification, digital video watermarking, and image mosaicing.