Indian Journal of Science and Technology Vol. 4 No. 9 (Sep 2011) ISSN: 0974- 6846

Adaptive search range decision for fast intra- and inter-prediction mode decision in H.264/AVC

Mehdi Jafari[1] and Shohreh Kasaei[2]
[1]Department of Electrical Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran
[2]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
mjafari@iauk.ac.ir; skasaei@sharif.edu

Abstract

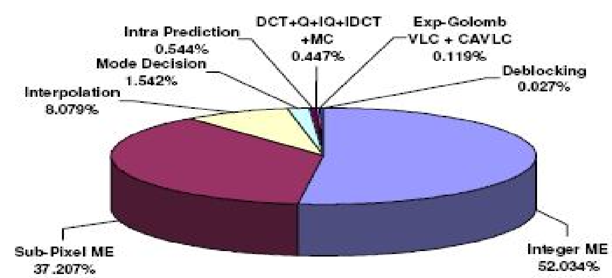
H.264/AVC, the latest video coding standard, adopts *rate-distortion optimization* (RDO) technique to obtain the best intra- and inter-prediction, while maximizing visual quality and minimizing the required bitrate. However, the full RD cost calculation for all intra-prediction modes, the exhaustive 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. The authors have previously proposed an approach for both inter- and intra-mode decisions that takes into account the several effective parameters, image content type, quantization parameter, correlation of motion vectors and motion details of video objects. In order to reduce the complexity, here we propose a new two-step adaptive search range decision by developing a combination of previous algorithm. The first step of our adaptive search range decision determines the search range for current MB according to the motion vectors of the previous MBs. In the second step, the motion vector obtained in the current frame is used to adaptively searching the window size of the previous reference frame for multi-frame motion estimation. 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 our previous algorithm and other algorithms reported in the literature.

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

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) (Wiegand *et al.,* 2003; Pan *et al.,* 2005). 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 (Wiegand *et al.,* 2003; Pan *et al.,* 2005). 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 previous standards. However, theses functionalities have made H.264/AVC difficult for applications with low computational capabilities. Thus, until now, the reduction of its complexity is a challenging task. The improvement in coding performance comes mainly from the intra- and inter-prediction part. H.264/AVC employs the Lagrange RDO method to achieve 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 the mode having the minimum required bitrate. The reference software of H.264/AVC, JM7.1, adopts a full search for both motion estimation and intra-prediction. The run-time percentage of each function is shown in Fig.1 (Huang *et al.,* 2006). As shown in this figure, 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.

Fig. 1. Run-time percentages of functional blocks in H.264/AVC baseline encoder.



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 (Changsung *et al.,* 2003). Thus, the computational burden of these types of brute force-searching algorithm is far more demanding

Indian Journal of Science and Technology Vol. 4 No. 9 (Sep 2011) ISSN: 0974- 6846

than any existing video coding algorithm.

The challenging topic of fast mode selection for intra- and inter-prediction is considered in this paper. To reduce the computational complexity, many algorithms have been proposed. For fast inter-prediction mode decision, the early termination technique (Tourapis *et al.,* 2003) reduces the number of potential prediction modes. Lim *et al.* (2003) proposed a classification method. Later, fast inter-mode selection algorithms were proposed (Kim *et al.,* 2004a,b; Martin *et al.,* 2004) to alleviate the encoder complexity due to the motion estimation and inter-mode decisions. Dai *et al.* (2004), presented a fast inter-prediction mode decision based on pre-encoding process.

Fast intra-mode decision algorithms using edge detection histogram and local edge detection are proposed by Pan *et al.* (2003, 2004, 2005). 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 (Kim *et al.,* 2005) with saving time of 70%, and statistical-based methods (Garg *et al.,* 2005) with saving time of 45%. Another fast intra-mode decision scheme is proposed by Jeon *et al.* (2003), 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 by Yang *et al.* (2004). This algorithm achieves 10% to 40% of computation reduction while maintaining similar PSNR and bitrate performance of H.264/AVC codes. Meng *et al.,* (2003) presented an *efficient intra-prediction* (EIP) algorithm based on early termination, selective computation of highly probable modes, and partial computation of the cost function. Also, an improved cost function to improve the coding performance is proposed byTseng *et al.* (2004; Bentao *et al.,* 2010;Tamtaoui *et al.*, 2010). Also Hsu *et al.* (2006) obtained a fast algorithm based on local edge information by calculating edge feature parameters.

Jafari and Kasaei (2008) proposed a new approach for both inter-and intra-mode size selection. For intra-prediction mode decision, we improved Pan's method (Pan *et al.*, 2003, 2004, 2005), and for inter-prediction mode decision, the split/merge procedure was used. Also, an effective method for accelerating the multiple reference frames ME without significant loss of video quality was proposed that was based on analyzing the available information obtained from previously processed frames.

This paper presents a two-step adaptive search range decision algorithm to improve the proposed algorithm (Jafari & Kasaei, 2008; 2010). In the first step, the proposed adaptive algorithm determines the search range for current MB according to MVs of previous MBs. In the second step, the length of MV obtained in the current reference frame is considered as the maximum size of the search window for motion estimation in the previous reference frame. We have verified the different parts of the proposed algorithm step by step by implementing it on the JM7.1 reference software and have compared its performance with previous 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.

Mode decision and RDO in H.264/AVC

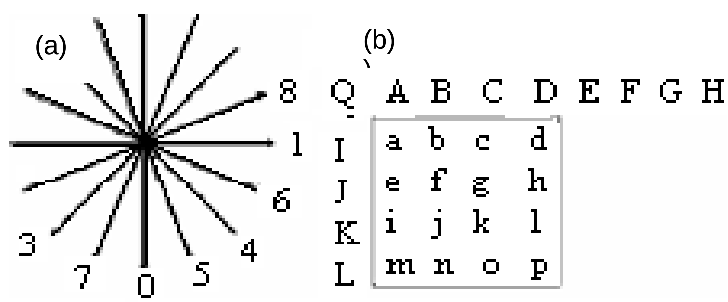
In common with earlier standards, H.264/AVC does not define the encoder, but defines the syntax of an encoded video bitstream together with the method of decoding the bitstream (Jafari & Kasaei, 2005; Jafari & Kasaei, 2006). The codec combines intra-picture prediction with inter-picture prediction to exploit the spatial and temporal redundancy, respectively.

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 Fig.2(a). In Fig. 2(b), the block (values of pixels "a" to "p") is to be predicted using A to Q pixel values.

Fig. 2 . (a) Intra-prediction modes for 4×4 luminance blocks. (b) Labeling of prediction samples.



The DC prediction (mode 2) is useful for those blocks with little or no local activities, the other modes (1-8) may only be used if all 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 (Richardson, 2005).

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).

Inter-prediction mode decision

Inter-prediction is based on using motion estimation and compensation to take advantage of 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).

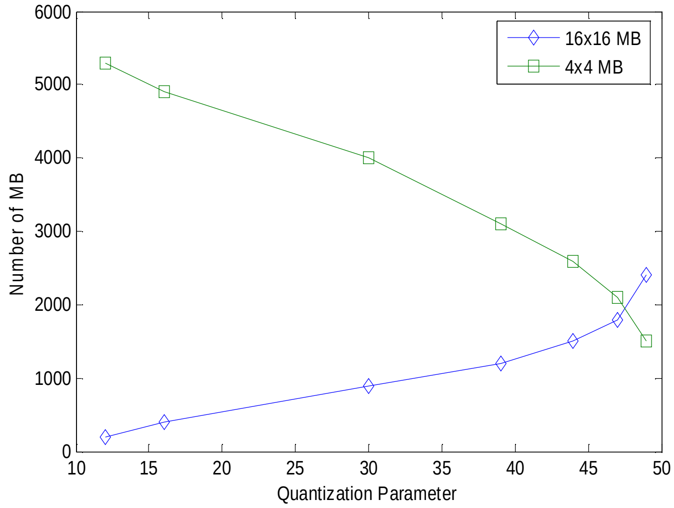
H.264/AVC as an enhanced reference picture selection as H.263++ enables efficient coding by allowing an encoder to select (for motion compensation purposes) among a large number of pictures that have been decoded and stored in the decoder.

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 has been reported (Changsung *et al.,* 2003). According to the procedure, 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(Changsung *et al*., 2003). Also, according to the RDO procedure of inter-prediction, for *M* block modes, *N* reference frames, and $\pm W$ search range, $M.N.(2W+1)^2$ positions should be tested for a single reference frame and a 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.

Proposed fast intra-prediction mode decision methods

Fig. 3. Number of 4×4 and 16×16 intra-coded macroblocks for different quantization parameters.



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: 1) Fig.3 shows the total number of 4×4 and 16×16 intra-coded macroblocks at different *quantization parameters* (QPs). As can be seen from this figure, fast detection of 4×4 intra-prediction mode can significantly improve the encoding speed at low QPs, while 16×16 intra-predictions can improve the speed at large QPs, 2) The prediction modes of each block are correlated with those of neighboring 4×4 luminance blocks, 3) Normally, pixels along the direction of local edges have similar values. Therefore, a good prediction can be achieved by predicting the pixels using the neighboring pixels that lie in the same edge directions, 4) The optimal mode (found by a full-search) and other "good" (second or third best) modes are most likely to have similar directions, 5) The directional features of 4×4 blocks can be preserved roughly after down-sampling and 6) Experimental results show that the reference pixels of a 4×4 luma block are likely to be similar to each other (Jafari & Kasaei, 2005).

Based on these observations, we have proposed a fast intra-prediction mode selection algorithm. In this section, some new ideas are combined with the fast mode selection algorithm introduced (Lim *et al*., 2003; Kim & Altunbasak, 2004; Pan *et al*., 2005) to improve their efficiency.

Improved pan's method for fast decision of I4MB

Pan *et al.* (2005) presented 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 the directional vector of each pixel in a block by

$$\vec{D}_{i,j} = \{dx_{i,j}, dy_{i,j}\} \quad (1)$$

Where the Sobel operator are defined by

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

$$dy_{i,j} = P_{i+1,j-1} + 2\times P_{i+1,j} + P_{i+1,j+1} - P_{i-1,j-1} - 2\times P_{i-1,j} - P_{i-1,j-} \quad (2)$$

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

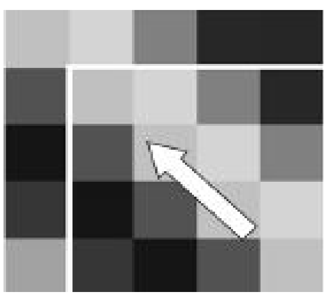
$$Amp(\vec{D}_{i,j}) = |dx_{i,j}| + |dy_{i,j}| \quad (3)$$

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

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

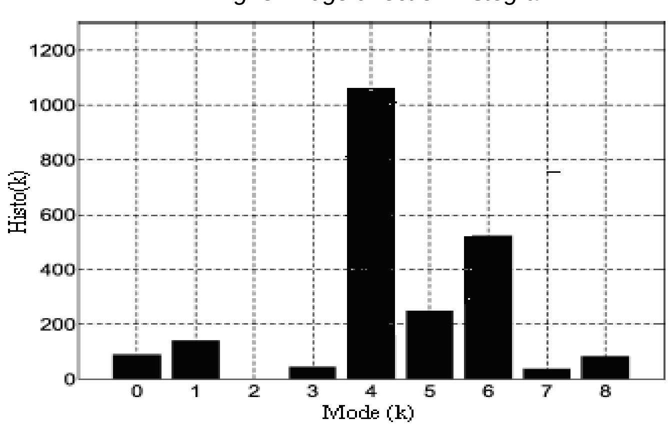
Then, the *edge direction histogram* (EDH) of the block is found (that indicates the number of pixels with similar edge directions). Therefore, the cell *k* with the maximum amount indicates that there is a strong edge along that direction in the block and thus is assigned as the dominant block direction. Fig.5 shows the EDH of the image shown in Fig.4.

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



In the Pan's method, for I4MB there are 4 modes (1 DC, 1 from maximum amplitude of EDH, and its 2 neighbors) with 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 will 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 computed to check whether it is smaller than a predetermined threshold or not, using

Fig. 5. Edge direction histogram



$$diff = \sum_{i=0}^{i=15} |avg - p_i| \quad (5)$$

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

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, carry out the RDO procedure for 3 modes at the most (M1 and its two neighbors).
3. Else, if *diff* <T, carry out the RDO procedure for two candidate modes at the most. The DC with maximum 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 computed by

$$diff1 = \sum_{i=0}^{i=64} |avg - p_i| \quad (6)$$

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

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 listed in Table 1. This table summarizes the number of candidates selected for RDO calculation based on EDH. 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 (592).

Table 1. 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

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 methods can achieve better results. The proposed algorithm is as follows:

1. Find the maximum value of EDH. 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. 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 9 prediction modes, but only one prediction mode is enough (Hsu et al., 2006).

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. 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 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 used the early termination of RDO calculations for all proposed algorithms as in (Pan *et al.,* 2005).

Proposed fast inter-prediction mode decision method

The motivated facts about the inter-prediction are: 1) 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, 2) It is observed that in natural image 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 and 3) Video background is not homogeneous, but because of temporal redundancy it is coded using 16×16 block sizes.

Fast inter-prediction mode decision using split and merge procedure

The split procedure partitions the MB into variable size blocks using a quad-tree approach. In this method, a macroblock 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:

Table 2. MB division

MV0	MV1
MV2	MV3

1. Subtract the current frame from its previous frame and for any 16×16 MB compute

$$N_i = number\ of\ pixels\ belong\ to\ the\ set\ MB\quad i$$

$$N_{im} = number\ of\ nonzero\ pixels\ in\ difference\quad MB$$

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 (vide Table 2), choose 8×16. Terminate.
6. If MV0=MV2 and MV1=MV3, choose 16×8. Terminate.
7. Repeat Steps 2 to 4 for each 8×8 blocks, except that sub-blocks are 4×4.
8. Terminate.

The mode selection methodology employed in this paper is as follows. 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.

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 to decide on the number of references frames (Huang *et al*., 2006).

1. For *QP*=20, *QP*=30 and *QP*=40, it can be seen that 65%, 79%, and 95%, of macroblocks need only one reference frame, respectively. Therefore, the block matching process should be proceeded from the nearest reference frame to the farthest reference frame.
2. Another interesting point is that low bitrate cases are more likely to have the best reference frames close to the current frame than higher bitrate cases.
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%, and 02.69%.

4. In H.264/AVC, the SKIP mode is a special case of P16 $\times$16. The percentages of SKIP macroblocks after searching one reference frame are 44.57%, 62.69%, and 79.14% for *QP*=20, 30, and 40, respectively.

This result shows that a large percent of MBs are coded as 16$\times$16 or skip mode that use only one reference frame, while for large QP this fact is amplified. According to these observations, in the following, we have listed 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 entropy coded. If we face the situation for which the transformed and quantized coefficients are very close to zero in the first reference frame, stop the matching process for the remaining frames.

Table 4. Different methods used in our experiments.

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
M6	Proposed Alg.	Proposed Alg.(Adaptive search range)	Proposed Alg.	Yes

2. Calculate the *sum of absolute transform difference* (SATD). If it is less than a threshold (*THSATD)*, stop the searching process.

3. If the best reference frame is the previous frame and the best motion vector is the same as that of the SKIP mode or 16$\times$16 mode and *QP* is larger than a threshod (*THQP)*, the multiple reference frames loop will be early terminated. The determination of *THQP* is empirically obtained in (Huang *et al*., 2006). In the proposed method 76%-96% of computations for searching unnecessary reference frames can be avoided.

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

Adaptive search range decision algorithm

In reference software JM7.1, the fixed search range window size equal to 16 is used for motion estimation. This full search range is inefficient and increases the computational complexity. Therefore, a two-step adaptive search range decision algorithms is presented. In the first step, we have adopted the length of MV obtained in the first reference frame as the maximum size of the search window for the motion estimation in the second reference frame. And, the length of MV in the second reference frame is selected as the maximum size of the search window in the third reference frame, and so forth. This step of the proposed algorithm reduces the SW size for 4 reference frames, and for the first reference frame a full search is applied. In the second step, an adaptive search range decision method is presented that determines the search window according to the MVs of previous MBs.

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

The primary search window size is selected as the average of MVs of the encoded MBs immediately on left, above, above the left, and above the right of current MB, computed as

SWR.x=(MVA.x+MVL.x+MVAR.x+MVAL.x)/4

SWR.y=(MVA.y+MVL.y+MVAR.y+MVAL.y)/4

where SWR.x, MVAx, MVL.x, MVAR.x, and MVL.x denote the search window, motion vector above, motion vector left, motion vector above the right, and motion vector above the left, in the x axis, respectively. Similarly y stands for the y axis. By increasing the search window size around the computed SW, the speed of motion estimation can be improved. The increment of the SW size is stopped when the block matching has not improved with respect to the primary computed SW size.

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 (Sullivan *et al.,* 2001). Experiments were carried out on the recommended sequences with various quantization parameters for IPPP... type. For IPPP... experiments, the total number of frames was 300 for each sequence, and the period of I-frame was 100. The used test platform was 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 + two step adaptive search range decisions) with other available approaches. To show the impact of different parts of the algorithm, these parts were added at different steps and the results were analyzed. Thus, the experiments were ordered in seven states as listed in Table 4.

Comparisons with the case of exhaustive search (RDO) were performed with respect to the change of average PSNR (ΔPSNR), the change of average data

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

		$\Delta PSNR \ (dB)$				
		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
	M6	-0.014	-0.080	-0.032	-0.016	-0.10
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
	M6	-0.01	-0.061	-0.013	-0.01	-0.03
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
	M6	-0.012	-0.012	-0.05	-0.05	-0.01
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
	M6	-0.01	-0.037	-0.1	-0.076	-0.071

bits (ΔBit), and the change of average encoding time (Δ Time), respectively.

The PSNR is derived from

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

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

$\Delta \ time$ be defined as

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

Also, the bitrate increase is defined as

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

A group of experiments were carried out on different sequences. The encoding bitrates, the PSNR values, and the time saving factor (as compared with the H.264 RDO method) for four test sequences with different quantization parameters are listed in Tables 5-6. 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.

Fig. 6, 7, and 8 show the examples of RD and the complexity curves of sequences "*Akiyo*" (class A),

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

		$\Delta Bit \ \%$				
		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
	M6	1.624	0.789	0.604	0.346	0.512
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
	M6	0.615	0.941	0.941	0.576	0.735
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
	M6	0.924	1.001	0.634	1.021	0.876
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
	M6	0.830	0.210	0.830	0.897	0.932

"*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 these figures, the RDO, Pan's method, improved Pan's method, and three forms of fast proposed methods (Fast intra + Fast inter + Fast multi reference frames) are compared (vide Table 4). Fig.9-11 show examples of classification results and 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.

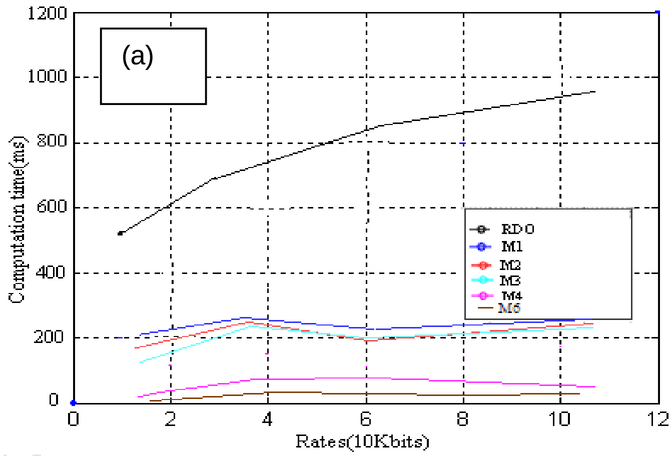
Conclusion

In this paper, an efficient intra- and inter-mode decision algorithm for H.264/AVC standard was integrated using a two-step adaptive search window decision. In the proposed algorithm, we decreased the encoding time by reducing the number of candidate modes. In order to achieve a better performance, adaptive search range for multiframe reference and MV of encoded MBs with some strength points were proposed to improve previous algorithms.

In order to evaluate the impact of different parts of the proposed algorithm, they were added step by step and the related experimental results were shown. The

Indian Journal of Science and Technology Vol. 4 No. 9 (Sep 2011) ISSN: 0974- 6846

Fig. 6. "Akiyo" sequence (IPPP seq.).
(a) Computational complexity.



(b) R-D performance.

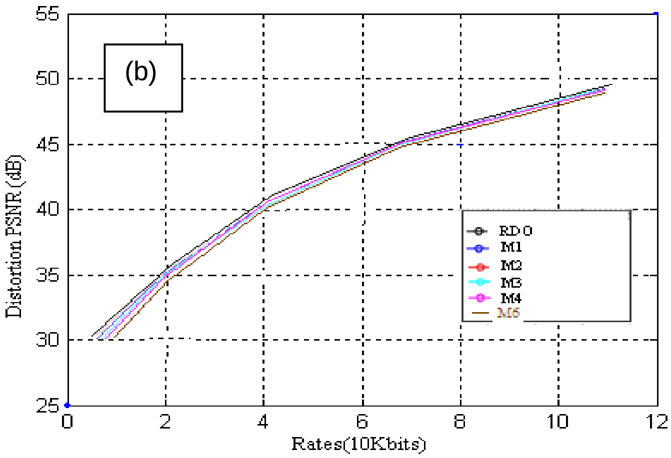
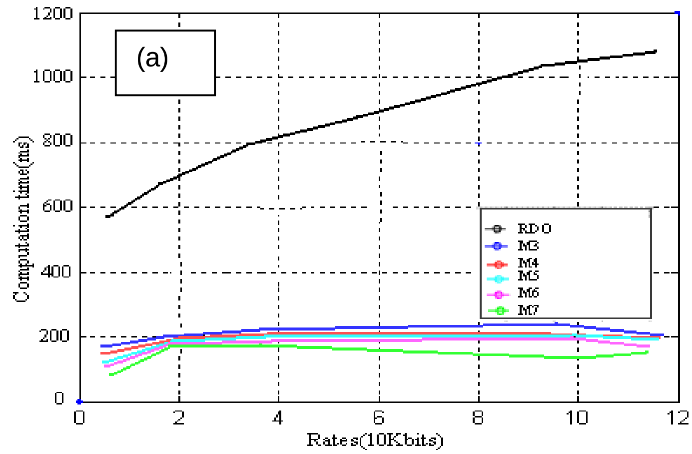


Fig. 7. "Foreman" sequence (IPPP seq.).
(a) Computational complexity.



(b) R-D performance.

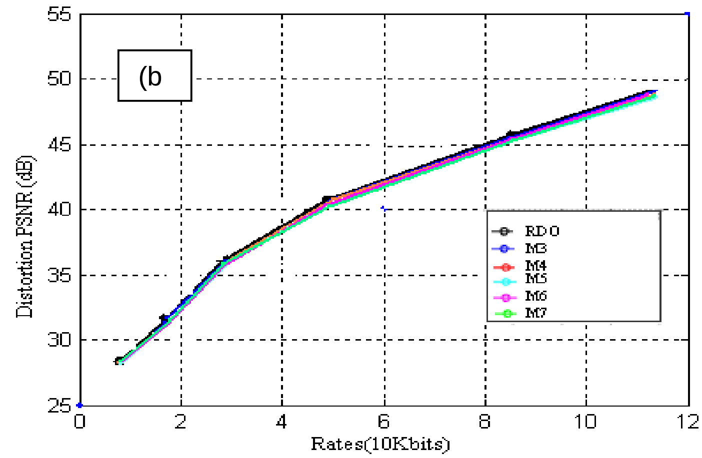
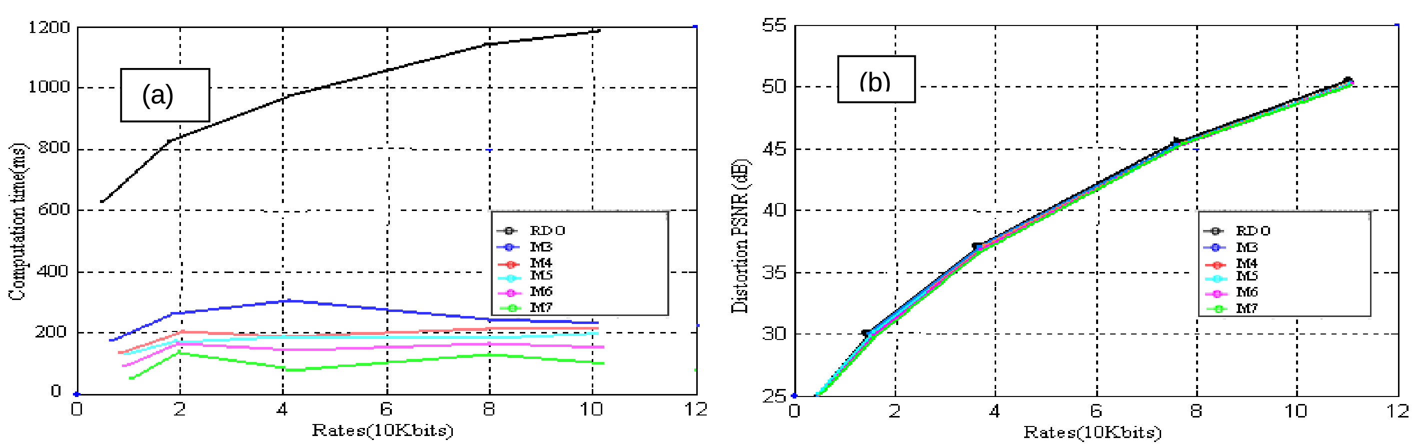


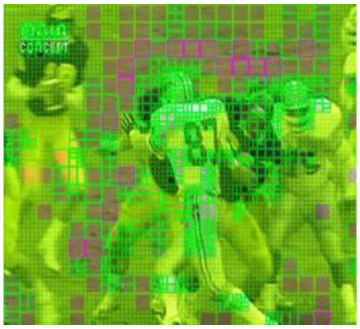
Fig. 8. "Stefan" sequence (IPPP seq.). (a) Computational complexity. (b) R-D performance.



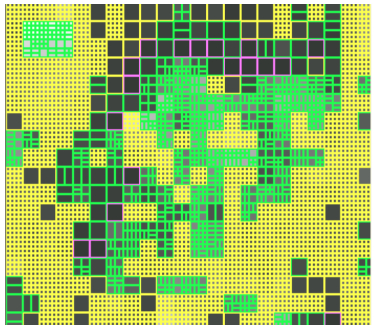
Indian Journal of Science and Technology Vol. 4 No. 9 (Sep 2011) ISSN: 0974- 6846

Fig. 9. Samples of "Football" sequence.

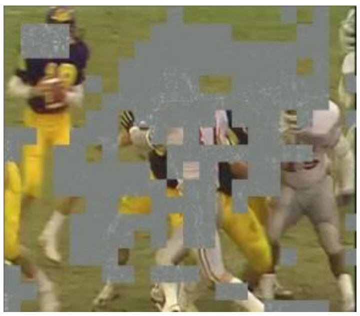
(a) MB division.



(b) MB division, without background.



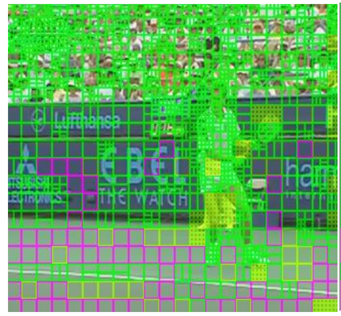
(c) Classification result of intra- and inter-prediction.



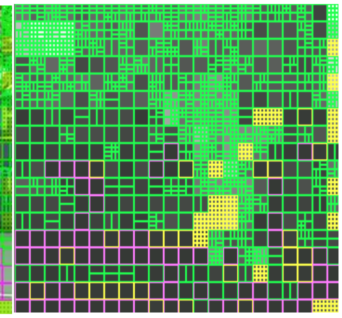
(d) Motion vector, and intra-prediction modes.

Fig. 10. Samples of "Stefan" sequence.

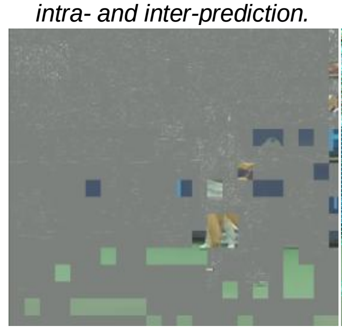
(a) MB vision.



(b) MB division, without background



(c) Classification result of intra- and inter-prediction.



(d) Motion vector, and intra-prediction modes.

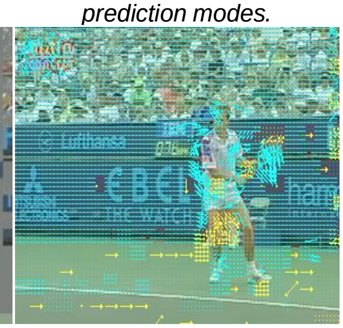
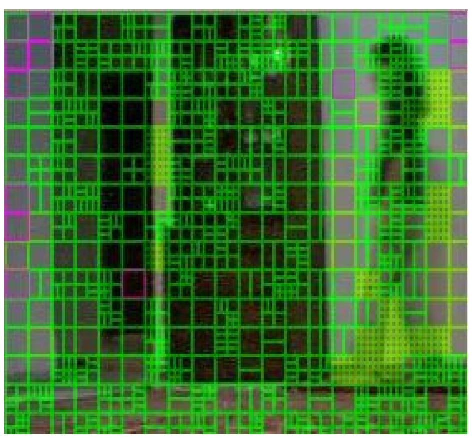
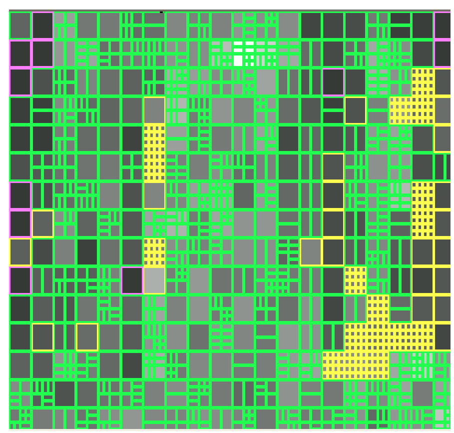


Fig.11.Samples of "walking person".

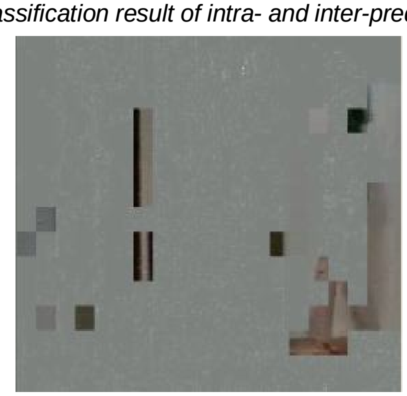
(a) MB division.



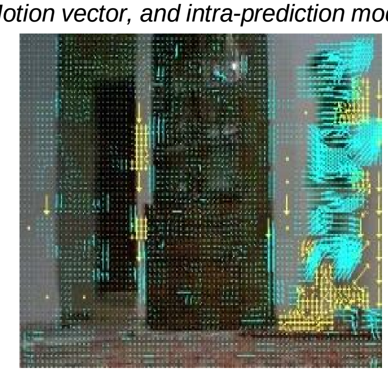
(b) MB division, without background.



(c) Classification result of intra- and inter-prediction.



(d) Motion vector, and intra-prediction modes.



experimental results showed that the proposed algorithm has reduced the number of RDO calculations with respect to the original and has improved the previous algorithms with a 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- and inter-prediction mode decision in the H.264/AVC video encoders with low computational cost.

References

1. Pan F, Lin X, Rahardja S, Lim KP, Li ZG, Wu D and Wu S (2005) Fast mode decision algorithm for intra-prediction in H.264/AVC video coding. *IEEE Trans. on Circuits & Systems for Video Tech.* 15(7), 813-822.
2. Wiegand T, Sullivan GJ, Bjontegaard G and Luthra A (2003) Overview of the H.264/AVC video coding standard. *IEEE Trans. on Circuits & Systems for Video Tech.* 13(7), 560-576.
3. Yu-wen Huang, Bing-Yu Hsieh, Shao-Yi Chien, Shyh-yih Ma and Liang-Gee Chen (2006) Analysis complexity reduction of multiple reference frames motion estimation in H.264/AVC. *IEEE Trans. on Circuits & Systems for Video Tech.* 16(4), 507-522.
4. Changsung Kim, Qing Li, Jay CC and Kuo (2003) Fast intra-prediction model selection for H.264 Codec. In: SPIE Intl. Symposium ITCOM 2003, Orlando, Florida, Sep. 7-11.
5. Lim KP, Wu S, Rahardja DJS, Lin X, Pan f and Li ZG (2003) Fast inter mode selection. 9th Joint Video Team (JVT) Mtg, San Diego, CA, USA.
6. Kim H and Altunbasak Y (2004a) Fast mode decision for inter prediction in H.264. In: ICIP 2004, Singapore.
7. Kim YH, Yoo JW, Lee SW and Shin J (2004b) Adaptive mode decision for H.264 encoder. In: *Electronic Lett*. 40(19).
8. Yu CA and Martin G (2004) Advanced block size selection algorithm for inter frame coding in H.264/MPEG-4 AVC. In: ICIP 2004, Singapore.
9. Dai Q, Zhu D and Ding R (2004) Fast mode decision for inter prediction in H.264. In: ICIP 2004, Singapore.
10. Pan F and Lin X (2003) Fast mode decision for intra-prediction, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6. In: JVT 7*th* Meeting, Pattaya II, Thailand.
11. Pan F, Lin X, Rahardja S, Lim KP and Li ZG (2004) A directional field based fast intra-mode decision algorithm for H.264 Video Coding. *IEEE Intl. Conf. on Multimedia & Expo*. 2, 1147-1150.
12. Kim J and Jeong J (2005) Fast intra-mode decision in H.264 video coding using simple directional masks. In: Proc. of SPIE, Vol.5960, pp:1071-1079.
13. Garg R, Jindal M and Chauhan M (2005) Statistics based fast intra-mode detection. In: Proc. of SPIE, Vol. 5960, pp: 2085-2091.
14. Jeon B and lee J (2003) Fast mode decision for H.264, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT. In: 10th Meeting, Waikoloa, Hawaii.
15. Yang C, Po L and Lam W (2004) A fast H.264 intra-prediction algorithm using macroblock properties. In: ICIP. pp: 461-464.
16. Meng B, Au OC, Wong C and Lam H (2003) Efficient intra-prediction algorithm in H.264. In: IEEE. pp: 837-840.
17. Tseng C, Wang H and Yang J (2004) Improved and fast algorithms for intra-4×4 mode decision in H.264/AVC. In: IEEE. pp: 2128-2131.
18. Hsu C, Ho M and Hong J (2006) An efficient algorithm for intra-prediction in H.264. IEEE. pp: 35-36.
19. Jafari M and Kasaei S (2008) Fast intra and inter prediction mode decision in H.264 advanced video coding. IJCSNS. 8(5), 130-140.
20. Jafari M and Kasaei S (2006) An efficient intra-prediction mode decision algorithm for H.263 to H.264 Transcoding. 10th IEEE Intl. Conf. on Computer Systems & Appl. pp: 1082-1089.
21. Jafari M and Kasaei S (2005) Prioritisation of data partitioned MPEG-4 video over GPRS/EGPRS mobile networks, *Asian Internet Engg. conf*. (AINTEC), Thailand. pp: 68-82.
22. Richardson (2005), H.264/MPEG-4 part 10 white paper, available: http://www.vcodex.fsnet.co.uk/resources.html.
23. Sullivan G (2001) 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.
24. Jafari M and Kasaei S (2010) Adaptive search range decision for fast intra- and inter-prediction mode decision in h.264/AVC. In: *The 2nd National Electrical Engg. Conf*. NEEC, 247-252.
25. A. Elyousfi, A. Tamtaoui, and E. Bouyakh (2010) A new fast intra prediction mode decision algorithm for H.264/AVC encoder. *Intl. J. Electrical & Electronics Engg*. 4(1), 89-95.
26. Bentao, Wu and Minyuan (2010) an improvement intra prediction algorithm for H.264. *J. Natural Sci*. 2010. 15(4), 340-344.