Data Compression Conference

A Novel Partial Prediction Algorithm for Fast 4x4 Intra Prediction Mode Decision in H.264/AVC

***Y N Sairam*[1], *Nan Ma*[1], *Neelu Sinha*[12]**
[1]*ATC Labs, NJ, USA*
[2] *Dept. of Computer Science & Math, Fairleigh Dickinson University, NJ, USA*

ABSTRACT

This paper proposes a partial prediction approach for fast mode decision in H.264/AVC 4x4 intra-prediction, exploiting the inherent symmetry existing in the spatial prediction modes. The high compression efficiency of H.264 comes with a price of high computational complexity. The intra-prediction process with its robust full intra mode search rate distortion optimization (RDO) technique to select the best mode for each macroblock is one such computational bottleneck. Unlike many fast algorithms which search for dominant edge in each block the existence of which is never assured, our algorithm quickly tries to find a closer prediction mode for each 4x4 block using a simple cost measure derived from the symmetry in all the prediction modes. The proposed algorithm only searches 3 or lesser modes instead of 9 for the 4x4 luma blocks. Experimental results reveal the significant computational savings achieved with slight PSNR degradation and bit-rate increase.

1. INTRODUCTION

The H.264/Advanced Video Coding (AVC) standard [1] uses intra-prediction to remove the spatial redundancy existing in a given frame/picture. The efficient and flexible implementation of intra-prediction using variable block sizes in H.264/AVC is one of the key algorithmic tools responsible for its excellent coding performance and quality at low bit-rates, even compared to many of current state of the art still image coding schemes. To achieve the highest coding efficiency, H.264/AVC encoder [2] employs a robust nonnormative technique called Lagrangian rate-distortion optimization (RDO) technique to select the coding mode and reference frame for each macroblock (MB). It also uses an exhaustive full search (FS), where it calculates the rate-distortion cost (RDcost) of every possible mode and chooses the mode having the minimum value as the best mode for a MB and this process is repeatedly carried out for all the possible modes [3] for a given MB. As a result the complexity and computational load increase drastically making the intra-prediction algorithmic block a dominant component of the overall encoder complexity besides the motion estimation algorithm. In addition, intra-prediction is computed for intra-frame as well as inter-frame to determine the block type. Therefore it is highly desirable to develop fast intra-prediction mode selection algorithm which will not only reduce complexity and computational load, but at the same time maintain comparable bit-rates and PSNR without any compromise to visual quality.

A number of algorithms [4-18] have been proposed which reduce the complexity and computational load by restricting the number of possible candidate modes and/or computing only partial costs. However, these approaches increase the bit-rate, or increase the complexity due to the pre-calculations needed, and/or, effect PSNR degradation heavily. In an attempt to overcome these shortcomings, we propose an algorithm, which exploits the pixel to pixel correlation, as well as, the inherent symmetry present in the spatial intra-prediction process. This proposed method,

1068-0314/08 $25.00 © 2008 IEEE
DOI 10.1109/DCC.2008.93

appropriately termed, the **Partial Intra-Prediction** process, diminishes the computational complexity while maintaining comparable bit-rates and PSNR. Further, this method also offers flexibility to trade off between quality and computational complexity as explained in Section 3.

This paper is organized as follows. In Section 2, we present an overview of the H.264/AVC Intra-Prediction process, followed by a discussion of the several different approaches for fast Intra-prediction in the available literature, as well as the performance considerations. In Section 3, we discuss our proposed Partial Intra-Prediction algorithm, followed by a discussion of the experimental results in Section 4. Finally, Section 5 contains our concluding remarks.

2. BACKGROUND

2.1. Overview of H.264/AVC 4x4 Intra-Prediction mode decision

The H.264/AVC standard exploits the spatial correlation between adjacent macroblocks/blocks for Intra prediction as explained below. Intra prediction is based on the fact that these frames do not depend on earlier or later frames for reconstruction. However, the previously encoded blocks from within the same frame may be used as reference for new blocks. The resulting picture is referred to as an I-picture. Fig.1 below shows the current block **X** to be predicted and its spatially adjacent blocks U, V, W which are the left, up, and, up-right encoded 4x4 blocks respectively.

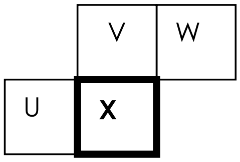
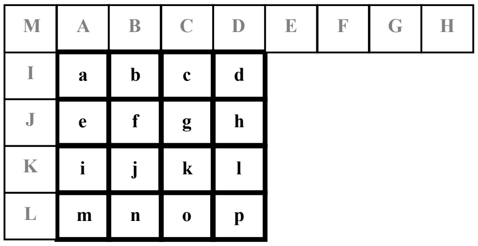


Fig.1: Adjacent blocks of current 4x4 block

In H.264/AVC Intra prediction is performed at two levels hierarchically. For coding the luma component, one 16x16 macro-block may be predicted as a whole using four Intra_16x16 modes, or each 4x4 sub-block of the 16x16 macro-block may be predicted individually using a total of nine possible Intra_4x4 modes; prediction at the level of 4x4 blocks is beneficial when the correlation is more localized. Intra prediction for the chroma component uses similar techniques as those for luma Intra_16x16 prediction. The different prediction modes generally correspond to different edge orientations, and, closer the current block **X** is to a prediction block **P**, more likely the current block has an edge oriented in that direction. This gives smaller residue which in turn needs fewer bits to code.



M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Fig. 2: A 4x4 block and its neighboring pixels

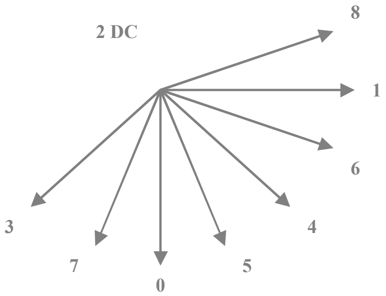


Fig. 3: Direction of 9 4x4 intra-prediction

Fig. 2 above shows a 4x4 block containing 16 pixels labeled from ***a*** through ***p***. A prediction block **P** is calculated based on the pixels labeled A-M obtained from the neighboring blocks. A prediction mode is a way to generate these 16 predictive pixel values using some or all of the neighboring pixels in nine different directions as shown in Fig. 3. Note that in some cases, not all of the samples A-M are available within the current slice. In order to preserve independent decoding of slices, only samples within the current slice are used for prediction. DC prediction is modified depending on which samples among A-M are available; the other modes may only be used if all of the required prediction samples are available (except that, if E, F, G and H are not available, their value is copied from sample D).

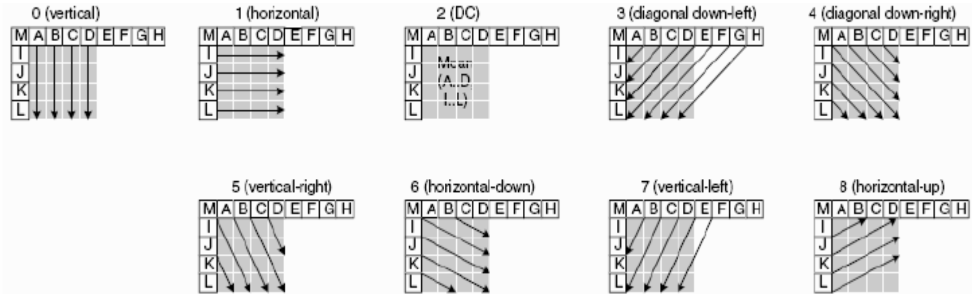


Fig. 4: Calculation of nine 4x4 intra-prediction modes

Fig. 4 shows how the nine intra prediction modes are designed in a directional manner. Mode 0 is the vertical prediction mode in which pixels a, e, i, and m are predicted by A and so on. Mode 1 is the horizontal prediction mode in which pixels a, b, c, and d are predicted by I and so on. Mode 2 is called DC prediction in which all pixels (a to p) are predicted by (A+B+C+D+I+J+K+L)/8. For modes 3-8, the predicted samples are formed from a weighted average of the prediction samples A-M as shown in figure. The encoder finally selects the best prediction mode for each block using rate-distortion optimization (RDOPT) that minimizes the residual between the encoded block **X** and its prediction **P**. The residue in all prediction modes is exactly transformed, quantized, entropy coded, and, then entropy decoded, de-quantized, inverse transformed as in conventional decoder and the decoded residue is used to obtain the original block **X'**. Now the sum of absolute differences between **X** and **X'** gives the RDcost. With Intra_4x4 prediction, the prediction error may be substantially smaller due to the small block size, although much higher number of numerical operations is required because of the larger number of blocks and the possibility of nine different directional predictions [3].

Since the choice of prediction modes for chroma components is independent to that of luma components, H.264/AVC encodes the MB by iterating all the luma intra decisions for each possible chroma intra prediction mode for the best coding efficiency. Therefore the number of mode combinations for luma and chroma components in an MB is $M8 \times (M4 \times 16+M16)$, where $M8$, $M4$, and $M16$ represent the number of modes for chroma, Intra_4x4 and Intra_16x16 predictions respectively. As a result, for an MB, it has to perform $4 \times (9 \times 16+4) =592$ different RDO calculations before a best RDO mode is determined which makes the intra-prediction process one of the computational bottlenecks in H.264/AVC encoder.

2.2. Current Approaches to fast Intra-Prediction

Broadly, the solution to the mentioned problem is attempted in two ways.

I. **Reduce candidate modes:** This approach tries to restrict the number of possible candidate modes by extracting the correlation characteristics of the current block with respect to its neighbor blocks through some efficient and simple computations [4-10].

II. **Partial Cost function:** This approach aims at arriving at a best mode through only partial computation of the complex RD cost function instead of whole [11-14].

There is yet another possible classification of fast intra-Prediction algorithms based upon the domain of operation.

2.2.1. Pixel Domain Techniques

Threshold based: In [4] [5], a threshold, based on neighboring costs, is used to achieve early termination in the cost computation of different modes, which makes it content dependent. A better attempt is made in [6], which overcomes the disadvantage of thresholds, though it could bring down the number of candidate modes to only six.

Local edge direction based: In [7] [8], an edge direction histogram is used, which introduces a lot of additional complexity in the form of pre-calculation cost. [9] [10] propose a simpler computation of the dominant edge direction within a block. However, the assumption of edge direction is not always true in every block. This leaves more candidate modes for further search when no dominant edge is present.

2.2.2. Frequency Domain Techniques

In contrast, frequency domain techniques are based on the inherent capability of Discrete Cosine Transform (DCT) bases to exploit the directional feature of the input block naturally. Since the DCT coefficients are anyway calculated this does not add any additional cost. [11] [12] cater to this argument though they differ in algorithms. However, a careful analysis and experimentation on DCT bases (basic tile patterns [13]) reveals that the above argument doesn't hold good for all directional predictions except the first few direct modes like 0, 1, 3 and 4. Also this fails completely if there is no dominant edge degrading PSNR badly. In [14] a complete transform domain cost function is presented. In [15-17] the correlation between optimal coding mode decisions of temporally adjacent pictures is exploited to reduce the computational load of the encoding, the computational saving is not good though.

The partial Cost computation is effective only in frequency domain as the DCT can shift the intensity of a whole block to the first few pixels. However, in [18], a very simple cost function is used to extract the local dominant edge direction.

3. PROPOSED ALGORITHM

A careful analysis of the intra prediction calculation process reveals that the main reasoning on which most of the above methods are based is some-what imprecise. These methods are based on the claim that the dominating direction of a smaller block is similar to that of the bigger macro-block. Another implicit assumption is that the directional correlation of each block is consistent with directions of the edges and that the prediction modes of each block are also correlated with those of neighboring modes. It is also interesting to note that most of these methods assume that the edge direction is very uniform. Thus, they pre-calculate the cost function for only a few modes like 0, 1, 3, and 4 and further evaluate directions like 5, 6, 7 and 8 if there is no clear distinction in those costs. Consequently, no method has any cost calculation for mode 2; it is either used all the time or is taken into account when the cost is almost distributed equally within the first four directions.

The algorithm proposed here overcomes all of the shortcomings of the above methods. Broadly speaking it is based on the *spatial symmetry* in the intra prediction process and uses pixel to pixel correlation as the basis for obtaining a subset of candidate modes in contrast to local edge correlation used in all the above methods. This way it doesn't assume a dominant edge in the block and has a separate and simple pre-calculation cost for all modes including mode 2. Additionally, this method also offers flexibility to trade off between quality and computational complexity whereby we can choose the number of candidate modes based on the PSNR degradation we are willing to accept. Effectively this method is termed as ***Partial Intra-prediction*** process.

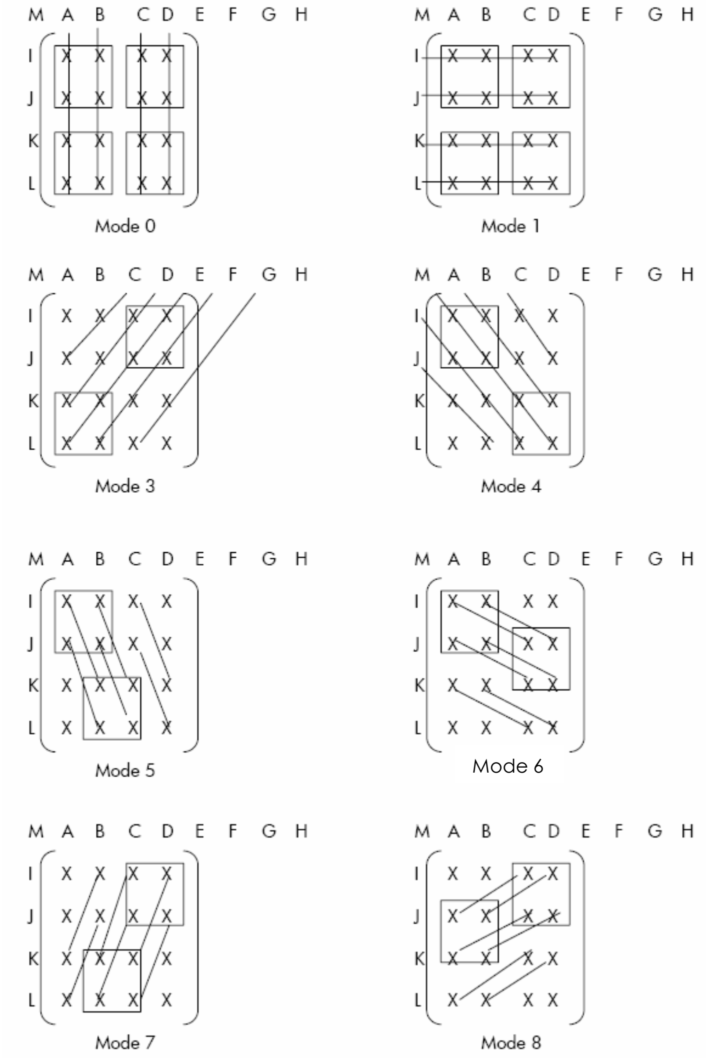
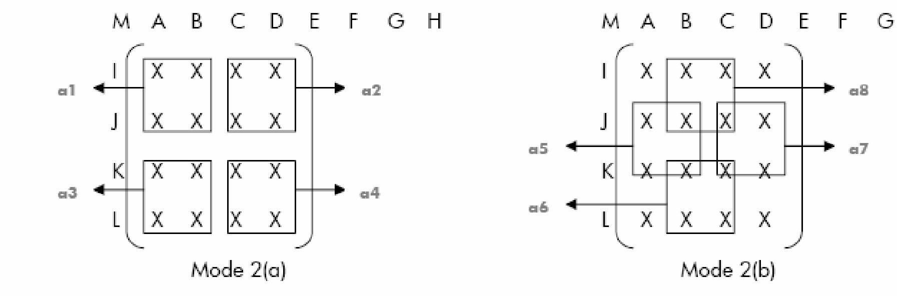

Mode 0
Mode 1
Mode 3
Mode 4
Mode 5
Mode 6
Mode 7
Mode 8

Fig. 5: Inherent Symmetry and Pixel Correlation in different intra-prediction modes



Mode 2(a)
Mode 2(b)

Fig. 6: 2x2 Sub-blocks

To fully understand the proposed algorithm we first look at the inherent symmetry of the Intra Prediction process. This symmetry is then utilized in developing low overhead cost measures which allow us to substantially reduce the range of possible modes for optimal prediction using Rate Distortion Optimization.

3.1. Inherent Symmetry in Intra-prediction Modes

Let us take a closer look at Fig. 5 which depicts the inherent symmetry in the various intra-prediction modes. Assume that this figure shows the prediction blocks of all modes of the current block indexed with respective mode numbers. From this representation the pixel to pixel correlation can be easily observed in each mode. For example, in Mode 0- the Vertical prediction mode, due to the symmetry, we can see that the pixels in each column will have the same predicted pixel value. Similarly, in Mode 1 – the Horizontal mode, the pixels in each row will have the same predicted value. Subsequently, such symmetry observations can be extended to all the different modes.

Now, consider eight 2x2 sub-blocks in the original 4x4 block X at different positions as shown in Fig.6. They are labeled as a_j where j is an index to position of sub-block. Let m_{ij} represent similar sub-block in i^{th} prediction mode at j^{th} position. We may consider the same notation for the sub-block average also. Thus, for Mode 0 with Vertical symmetry, we can see that m_{01} is same as m_{03}, and, m_{02} is same as m_{04}. Similar symmetry exists in all modes as summarized in Table 1 below.

Table 1: Inherent Symmetry in Intra-prediction modes

Mode	Symmetry	Sub-Block Cost Measures
0	$m_{01}=m_{03}$; $m_{02}=m_{04}$ or $m_{08}=m_{06}$	$\lvert a_1 - m_{01}\rvert + \lvert a_2 - m_{02}\rvert + \lvert a_3 - m_{01}\rvert + \lvert a_4 - m_{02}\rvert$
1	$m_{11}=m_{12}$; $m_{13}=m_{14}$ or $m_{15}=m_{17}$	$\lvert a_1 - m_{11}\rvert + \lvert a_3 - m_{12}\rvert + \lvert a_2 - m_{11}\rvert + \lvert a_4 - m_{12}\rvert$
2	$m_{21}=m_{22}=m_{23}=m_{24}$ or $m_{25}=m_{26}=m_{27}=m_{28}$	$\lvert a_1 - m_{21}\rvert + \lvert a_2 - m_{21}\rvert + \lvert a_3 - m_{21}\rvert + \lvert a_4 - m_{21}\rvert$
3	$m_{32}=m_{33}$	$(\lvert a_2 - m_{31}\rvert + \lvert a_3 - m_{31}\rvert) x 2$
4	$m_{41}=m_{44}$	$(\lvert a_1 - m_{41}\rvert + \lvert a_4 - m_{41}\rvert) x 2$
5	$m_{51}=m_{56}$ or $m_{58}=m_{54}$	$\lvert a_1 - m_{51}\rvert + \lvert a_6 - m_{51}\rvert + \lvert a_8 - m_{52}\rvert + \lvert a_4 - m_{52}\rvert$
6	$m_{61}=m_{67}$ or $m_{65}=m_{64}$	$\lvert a_1 - m_{61}\rvert + \lvert a_7 - m_{61}\rvert + \lvert a_5 - m_{62}\rvert + \lvert a_4 - m_{62}\rvert$
7	$m_{72}=m_{76}$ or $m_{78}=m_{73}$	$\lvert a_2 - m_{71}\rvert + \lvert a_6 - m_{71}\rvert + \lvert a_8 – m_{72}\rvert + \lvert a_3 - m_{72}\rvert$
8	$m_{85}=m_{82}$ or $m_{83}=m_{87}$	$\lvert a_5 - m_{81}\rvert + \lvert a_2 - m_{81}\rvert + \lvert a_3 - m_{82}\rvert + \lvert a_7 - m_{82}\rvert$

3.2. Spatial domain feature

We use spatial domain feature to filter out unlikely mode candidates. For each prediction mode, we can compute the sum of absolute differences (SAD) between the true and predicted pixel values as a spatial domain feature.

3.3. Partial Prediction Process

Using the above symmetry in different prediction modes, from Table 1, we now try to simplify the intra prediction process as follows. We try to obtain a close measure to SAD cost function in each mode (which is just the sum of absolute differences between16 pixels in X and P) in terms of those sub-blocks which come under symmetry in respective prediction modes as listed in the Table 1 above.

In mode 0, the sum of absolute differences between a_1 and m_{01}; a_2 and m_{02}; a_3 and m_{03}; a_4 and m_{04} gives the cost. However since $m_{01}=m_{03}$ and $m_{02}=m_{04}$ in mode0 due to the symmetry we can simply calculate the cost function as $|a_1 - m_{01}| + |a_2 - m_{02}| + |a_3 - m_{01}| + |a_4 - m_{02}|$, where $|x|$ stands for the absolute value for x. Similarly cost is calculated in all other modes exploiting the existing symmetry in corresponding prediction mode as listed out in Table 2. The pixels which are not covered under symmetry (like in modes 3-8) are taken care of by replicating the pixel differences that are covered under symmetry. The fact that the intensity variations within a block are not so rapid from pixel to pixel and also that these neighboring pixels are similarly predicted from the same adjacent pixels justifies the assumption.

3.4. Pre-Calculation Costs

Note that all of the m_{ij} measures can be calculated from 13 pixels A-M from neighboring blocks using the corresponding pixel values in respective positions in each prediction block depending on the mode. The cost function is simplified by using averages of sub-blocks instead of pixels directly. It is further simplified by the symmetry observed in table 1, which brings down the number of 'm' measures that are to be pre-computed in each mode. As an example mode 0 requires only m_{01} and m_{02} to be computed as (A+B)/2 and (C+D)/2 respectively instead of 4 measures $m_{01}$to m_{04}. Modes 2-4 require only one such measure to be computed. And rest of the modes require atmost two such measures. So a total of 15 measures are required to be computed in addition to 8 averages a_j in the original block to calculate the cost function for all modes. This is the total pre-calculation cost for a block. To further cut down the pre-calculation cost, we avoid divisions by considering only additions of pixels in sub-blocks rather than averages. This final pre-calculation cost is insignificant compared to the computational gain achieved as discussed below.

3.5. Significant and Flexible configuration : Quality vs. Complexity Tradeoffs

The prediction mode giving the least cost can ideally be chosen as the best mode. However to pay some price for our simplification without much loss in visual quality we may take first few least cost modes in increasing order and obtain the best mode using the RDO process. Our experimental results in section 4 suggest '3' to be the best selection for candidate modes (mode_count) that can give a good compromise between computational saving and PSNR loss. This parameter '*mode_count*' gives us the control to trade-off between gain and loss efficiently. We may also use a threshold to the cost differences in table 2 to choose the number of candidate modes, which will give a non-integer value for average mode_count. Even a mode_count of 3 will reduce the total RDO calculations to as low as 4 x (3 x 16 + 4) =208, which is just about 35% of total computations. This is a computational saving by 65% ideally.

3.6. Criteria for further reducing 'mode_count'

In [19], authors present a detailed analysis of mode filtering using SAD. We use two of the reported results to further reduce the mode_count from 3 to 1.

1. If the smallest SAD cost value is lower than 50, we may directly use the least mode with the probability of error lower than 10%. And the cumulative histogram shows that approximately 38% of blocks fall in to this event. The error probability is defined as

$$\Pr ob(e \mid f(m_{SAD})) = \Pr ob(m_{SAD} \neq m_{RDO} \mid f(m_{SAD})) \tag{1}$$

Where m_{SAD} and m_{RDO} are modes selected based on optimal SAD and RDO criteria respectively and $f(m_{SAD})$ represents the SAD value of mode m_{SAD}.

2. Further we may consider higher modes if the corresponding mode cost is below a certain threshold. This is formulated as below using a parameter T.

$$\Pr ob(e \mid T) = \Pr ob(m_{SAD} \neq m_{RDO} \mid T = \frac{SAD_2 - SAD_1}{SAD_1}) \tag{2}$$

For T larger than 38%, we can skip next modes with error probability less than 10%.

3.7. Summary of the Proposed Algorithm:

Let the input block be X and A to M be the pixels from neighboring blocks available for prediction.

1. Calculate sub-block averages a_i for current block X as in Fig. 6.

2. Calculate averages m_{ij} in each mode from A-M pixels for those sub-blocks covered under symmetry as listed in Table 1.
3. Calculate the cost measure of each mode as shown in Table 1.
4. Sort and pick up three or lesser candidate modes giving the least cost in ascending order for further evaluation.
   a. If least SAD is less than 50, choose only least mode; mode_count=1.
   b. Further include second and third modes only if corresponding parameters (T) exceed 0.38 as in 3.6. This will give a mode_count of 2 or 3 at most.
   c. The most_probable_mode may also be added to the candidate modes list if it was not already selected through pre-calculations.
5. Now do the exact RDO evaluation for these few candidate modes to obtain the best prediction mode for the macroblock.

3.8. Advantages of this approach:

1. Significant computational saving (ideally 65%) with minimal PSNR loss.
2. Pre-calculation involves only additions and subtractions unlike many of earlier methods which used complex calculations for edge detection.
3. Separate pre-calculation cost for mode 2 which no earlier method has.
4. No assumption of dominant edge in blocks. Instead of edge correlation, it directly aims at pixel correlation. Hence overcomes many of previous shortcomings.
5. Flexibility to trade-off between computational gain and PSNR loss. In addition, the most probable mode predicted from the modes of adjacent already predicted modes may be added to candidate modes list if that mode is not already present.

4. EXPERIMENTAL RESULTS

The proposed method is first evaluated in Matlab through computer simulations in comparison to reference FS H.264/AVC intra prediction with SAD cost function. Standard gray scale test images like peppers, lena, goldhill, house, bird etc are used for our simulations. Results are shown in Table 4.1 for the case of mode_count exactly set to 3 and lesser than 3. The results show significant improvement compared to fast three step method [6] and DCT coefficients method [12]. Both the methods degrade PSNR badly almost by 0.5 to 1 dB for a little saving compared to SAD. The proposed method achieves better match to SAD compared to both of them with much lesser candidates and minimal PSNR loss.

Table 4.1: Partial Prediction Approach applied to test images

Images	FS with SAD		Proposed (mode_count = 3)			Proposed (mode_count < 3)		
	Modes	PSNR	Modes	% Match	dB loss	MODES	% MATCH	dB loss
peppers.png	9	26.9100	3	91.5527	0.0241	2.6768	90.6250	0.1753
house.png	9	29.9157	3	81.4453	0.0606	2.7942	81.2500	0.0902
bird.gif	9	30.7278	3	82.0313	0.0302	2.8647	81.8604	0.0553
camera.gif	9	24.4249	3	84.4238	0.0705	2.7505	83.2764	0.3193
goldhill.gif	9	25.1679	3	87.5488	0.0837	2.7292	86.8164	0.1291
lena.gif	9	26.2605	3	89.7949	0.0626	2.6523	89.1846	0.1160

The proposed algorithm is then implemented in JM10.1 provided by [2]. In our experiments RD Optimization is enabled, hadamard transform is used, initial QP used is 28 and intra period is set to 1. 16x16 Intra prediction is also disabled. The test sequences include Foreman, Container, walk, news, salesman in format of QCIF (4:2:0) and Foreman, Bus, Flower, Mobile, and Waterfall in format of CIF (4:2:0). The length of each test sequence is 300 frames and all the frames are encoded using I-frame coding. Tables (4.3 – 4.4) list the simulation results of the proposed algorithm compared to full search in terms of three measures: PSNR loss in dB(**ΔdB**), bit-rate (**ΔB(%)**) and computation time (**ΔT(%)**) changes as a percentage. The simulations are run on different sequences under the following four test conditions.

Proposed 1: with mode_count=3
Proposed 2: with mode_count<3, filtering using 4(b) in 3.7.
Proposed 3: with mode_count<3, filtering using 4(a) in 3.7.
Proposed 4: with mode_count<3, filtering using both.

Table 4.2: Partial Prediction Algorithm in JM 10.1 with QCIF sequences

Sequence	Proposed 1			Proposed 2			Proposed 3			Proposed 4		
	ΔT(%)	ΔdB	ΔB(%)	ΔT(%)	ΔdB	ΔB(%)	ΔT(%)	ΔdB	ΔB(%)	ΔT(%)	ΔdB	ΔB(%)
foreman	-49.92	-0.04	+2.18	-57.78	-0.1	+6.32	-59.28	-0.14	+8.43	-62.93	-0.17	+9.72
container	-49.62	-0.06	+3.85	-56.44	-0.11	+6.63	-58.84	-0.13	+9.4	-62.45	-0.16	+9.37
news	-49.89	-0.08	+4.08	-56.86	-0.14	+6.73	-58.4	-0.16	+8.42	-62.08	-0.2	+9.77
saleman	-49.84	-0.09	+3.07	-57.06	-0.14	+6.394	-58.06	-0.17	+8.64	-61.2	-0.2	+9.56
walk	-49.9	-0.07	+3.32	-56.76	-0.14	+6.078	-57.14	-0.15	+7.55	-61.382	-0.19	+8.77

4.1. Partial Prediction Algorithm in JM 10.1 with CIF sequences

Sequences	Proposed1 (mode_count=3)			Proposed4(Mode_count < 3)		
	ΔT(%)	ΔdB	ΔB (%)	ΔT(%)	ΔdB	ΔB (%)
foreman	-49.89	-0.04	+4.37	-62.72	-0.15	+9.26
bus	-51.23	-0.09	+2.89	-61.74	-0.18	+7.84
mobile	-51.34	-0.14	+1.88	-60.76	-0.22	+4.58
flower	-50.8	-0.12	+1.91	-60.04	-0.24	+4.25
waterfall	-49.87	-0.1	+2.34	-58.7	-0.19	+6.41

The performance of our proposed algorithm is obvious from the presented results. The number of candidate modes is reduced to the least compared to many of the fast mode decision methods [4-7]. With three least modes selected on average, our method achieves about 50% reductions in complexity with minimum PSNR loss of 0.06db on average. The savings increased to 62% when further mode filtering is done with atmost 0.2 dB loss in PSNR. [4-7] reported savings less than 35%. Very little computation time is lost in precalculations (5-15%) compared to most of edge direction histogram methods [8-10] which consume considerable amount of total computations. Also the argument that the presence of edge is not always guaranteed in every block explains the significant loss in PSNR above 0.2dB in these methods apart from slower speedup. The DCT domain techniques [11, 12] also significantly loose on quality due to the same reason. The Matlab results measure the performance match of the proposed method with FS reference method using SAD cost function. It is always above 80% with an average PSNR loss of about 0.06 dB for mode_count = 3. Overall, the proposed method gave better results compared to many earlier methods with significant computational saving, minimal PSNR loss and bit-rate increment. The proposed method gives us the advantage of complete control to trade-off between quality and computational saving.

5. CONCLUSIONS

In this paper, a fast intra-prediction mode decision method for H.264/AVC is proposed based on the pixel to pixel correlation and the symmetry inherently existing in the intra-prediction process, in contrast to the existing work which is based on the local edge correlation to prediction mode direction assuming that each block carries a dominant edge. So the proposed approach overcomes the shortcomings of the earlier methods and gives much better quality for given bit-rates with minimum PSNR loss as low as 0.05 db. It also therefore avoids the relatively complex computations required for extracting the dominant edge like in most of earlier methods and limits the pre-calculations to a set of additions and subtractions. There is significant speed-up (50%) overall while maintaining similar bit-rate (3.5%) and PSNR (~0.1dB). The results verify the success of the proposed method. Moreover the approach offers flexibility to the user to trade-off between quality and computational complexity.

6. REFERENCES

1. *Information Technology - Coding of Audio-Visual Objects - Part 10: Advanced Video Coding*, Final Draft International Standard, ISO/IEC FDIS 14496-10, March 2005.
2. JVT Reference Software JM10.1, http://iphome.hhi.de/suehring/tml/
3. Iain E. G. Richardson, H.264 and MPEG-4 Video Compression – Video Coding for Next-generation Multimedia, John Wiley & Sons, ISBN: 978-0-470-84837-1, 2003.
4. B. Meng, and O.C. Au, "Fast Intra-prediction Mode Selection for 4x4 blocks in H.264," *Proc. of 2003 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP03,* April 2003.
5. B. Meng, O.C. Au, C-W. Wong and H-K. Lam, "Efficient Intra-prediction Mode Selection for 4x4 blocks in H.264," *Proc. of IEEE Int. Conf. on Multimedia and Expo, ICME03,* vol. 3, July 2003.
6. C-C. Cheng and T-S. Chang, "Fast Three Step Intra Prediction Algorithm for 4x4 blocks in H.264," *Proc. of IEEE International Symposium on Circuits and Systems ISCAS 2005*, May 2005.
7. F. Pan, et al., "Fast Mode Decision for Intra Prediction," *JVT-G013, 7th Meeting,* Pattaya II, Thailand, 7-14 March, 2003.
8. F. Pan, et al., "A Directional Field Based Fast Intra Mode Decision Algorithm for H.264 Video Coding," *Proc. of 2004 IEEE Int. Conf. Multimedia & Expo, ICME04*, 2004.
9. Y-D. Zhang, F. Dai, and S-X. Lin, "Fast 4x4 Intra-prediction Mode Selection for H.264," *Proc. of 2004 IEEE Int. Conf. Multimedia & Expo, ICME04*, 2004.
10. J-F. Wang, et al., "A Novel Fast Algorithm for Intra Mode Decision in H.264/AVC Encoders," *Proc. of 2006 IEEE Int. Symposium on Circuits and Systems, ISCAS 2006,* May 2006.
11. T. Tsukuba, et al., "H.264 Fast Intra-prediction Mode Decision Based on Frequency Characteristic," *Proc. of European Signal Processing Conf. EUSIPCO2005*, 2005.
12. T. Hattori and K. Ichige, "Intra-prediction Mode Decision in H.264/AVC using DCT Coefficients," *Proc. Int. Symposium on Intelligent Signal Processing & Communication Systems, ISPACS 2006*, Dec 2006.
13. B. Shen and I.K. Sethi, "Direct Feature Extraction from Compressed Images," *Proc. IS&T/SPIE Conf. Storage and Retrieval for Image and Video Databases*, Jan 1996.
14. J. Xin, A. Vetro and H. Sun, "Efficient Macroblock Coding-mode Decision for H.264/AVC Video Coding," *Proc. of the 24th Picture Coding Symposium PCS '04*, December 2004.
15. X. Lu, et al., "Fast Mode Decision and Motion Estimation for H.264 with a Focus on MPEG-2/H.264 Transcoding," *Proc. of 2005 IEEE International Symposium on Circuits and Systems ISCAS 2005*, vol. 2, pp. 1246–1249, Kobe, Japan, May 2005.
16. M-C. Hwang, et al., "Fast Intra Prediction Mode Selection Scheme Using Temporal Correlation in H.264," *Proc. IEEE TENCON 05, Region 10,* Nov 2005.
17. C-C. Lien and C-P. Yu, "A Fast Mode Decision Method for H.264/AVC Using the Spatial-Temporal Prediction Scheme," *Proc. of 18th IEEE Int. Conf. on Pattern Recognition ICPR '06*, Volume 04, pp. 334-337, August 2006.
18. A Elyousfi, et al., "Fast Mode Decision Algorithm for Intra prediction in H.264/AVC Video Coding," ICJSN S Int. Journal of Computer Sc. and Network Security, Vol. 7, No. 1, Jan 2007.
19. C Kim, H Shih, C J Kuo, "Fast H.264 Intra prediction Mode selection using Joint spatial and Transform domain features", Elsevier Science, 25 April 2004.