Low Complexity Intra MB Encoding in AVC/H.264

Rashad Jillani *Student Member*, IEEE and Hari Kalva, *Senior Member*, IEEE

Abstract —*In this paper we introduce and evaluate a novel machine learning based approach to reduce the complexity of Intra macroblock (MB) coding. The proposed approach is based on the hypothesis that MB coding mode decisions in H.264/AVC video have a correlation with the intensities of adjacent MBs and sub-MBs. This paper also discusses and analyzes different approaches of using machine learning in Intra prediction. We discuss, amongst other features, slices, Intra prediction scheme for H.264 and data mining. We use data mining algorithms to develop decision trees for H.264 coding mode decisions. The proposed approach reduces the H.264/AVC MB mode computation process into a decision tree lookup with very low complexity. The proposed algorithm is implemented in reference software by modifying the source code and is compared with the JM reference software for H.264/AVC*[1].

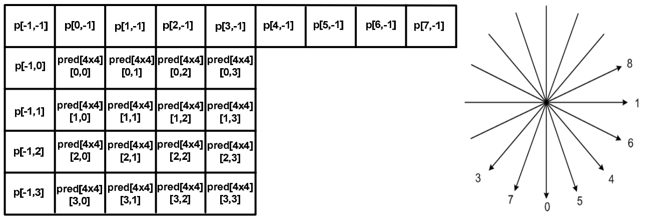
Index Terms — H.264/AVC, intra prediction, data mining, machine learning

I. INTRODUCTION

H.264/AVC is the latest video coding standard approved by ITU-T as Recommendation H.264 and by ISO/IEC as International Standard 14496-10 (MPEG-4 part 10) Advanced Video Coding (AVC) [1]. The compression efficiency of H.264/AVC has increased mainly because of the large number of coding options available. For example, the H.264 video supports Intra prediction with 3 different block sizes and Inter prediction with 8 different block sizes. The encoding of a macro block (MB) involves evaluating all the possible coding options and selecting an option that has the least cost associated with it. Resource constrained devices typically manage the complexity by using a subset of possible coding modes thereby sacrificing video quality. This quality and complexity relationship is evident in most video codecs used today. Most H.264 encoder implementations on mobile devices today do not implement the standard profiles fully due to high complexity.

Most of the traditional complexity reduction approaches in video coding are based on eliminating a subset of allowed coding modes and sacrifice quality for reduced complexity. The traditional approaches have not been able to reduce the encoding complexity enough to enable the use of advanced video coding features on resource constrained devices. We develop machine learning based approaches to reduce the complexity of video encoding. This approach reduces the computationally expensive elements of encoding such as coding-mode evaluation to a classification problem with negligible complexity. The key contribution of this work is the exploration of machine learning in video encoding applications. In this paper, we also focus our attention on macroblock mode decision, one of the most stringent tasks involved in the encoding process.

The H.264/AVC video coding standard uses the notion of macroblock (MB) partition to refer to the group of pixels in an area that share a common prediction. The encoder selects the coding modes for the macroblock, including the best macroblock partition and mode of prediction for each macroblock partition, such that the video coding performance is optimized. Intra prediction is one of new techniques, which is more complex and robust than the AC/DC prediction used in MPEG-4. Prior to H.264/AVC Intra prediction, most image coding standards utilize various differential pulse code modulation (DPCM) methods to achieve compression efficiency. In H.264/AVC, line-based Intra prediction (LIP) is used. Pixel values from the neighbors of the current block are extrapolated to construct a prediction block. Different directional predictions are defined by modes and used to exploit the spatial correlation that may exist between the predicted and actual pixels in a target block. Intra prediction in H.264/AVC predicts each pixel in the block from the previously encoded and reconstructed pixels in the nearby blocks. For the luma components, two Intra prediction modes are supported: Intra 4×4 prediction mode and Intra 16×16 prediction mode. The chroma prediction is similar to Intra 16×16 prediction in luma except for its different block size (8×8). In Intra 4×4 prediction mode, each 4×4 luma block can select one of nine prediction modes, which are shown in Fig. 1.



p[-1,-1]	p[0,-1]	p[1,-1]	p[2,-1]	p[3,-1]	p[4,-1]	p[5,-1]	p[6,-1]	p[7,-1]
p[-1,0]	pred[4x4] [0,0]	pred[4x4] [0,1]	pred[4x4] [0,2]	pred[4x4] [0,3]				
p[-1,1]	pred[4x4] [1,0]	pred[4x4] [1,1]	pred[4x4] [1,2]	pred[4x4] [1,3]				
p[-1,2]	pred[4x4] [2,0]	pred[4x4] [2,1]	pred[4x4] [2,2]	pred[4x4] [2,3]				
p[-1,3]	pred[4x4] [3,0]	pred[4x4] [3,1]	pred[4x4] [3,2]	pred[4x4] [3,3]				

Fig. 1. Intra 4x4 MB pixels with predicted boundary pixels and Intra 4x4 prediction directions

The 13 boundary pixels from previously coded blocks are used to generate the predicted image [5]. As an alternative to Intra 4x4 prediction, the entire 16x16 luma component of a macroblock may be predicted in one operation. And each 16x16 luma block can select one of four prediction modes. Among the prediction modes, the best mode is chosen and is subtracted from the current block to generate a residue for entropy coding. This residue is quantized, transformed,

---

[1] This work was supported in part by the FAU Research Corporation's Gap Fund Award.

The authors are with the Department of Computer Science and Engineering,777 Glades RD,33431 Boca Raton, FL, USA; Florida Atlantic University (rjillani@fau.edu, hari@cse.fau.edu).

Manuscript received January 15, 2009

inverse quantized, inverse transformed, and then added to the best prediction mode to generate a reconstructed image. The pixels in the reconstructed image are used to form the prediction mode of the current block [2]. Therefore, the Intra prediction process has to wait until reconstruction of the previous blocks is complete, and this dependency increases the computational complexity and makes it difficult to implement at real-time, especially for the video encoder in HDTV applications and also for the decoders in mobile devices. However, since the demand for the high definition video streams is growing, a low complexity and high speed Intra prediction scheme needs to be developed.

To take the full advantage of these modes and to select best of the available modes, H.264/AVC encoder can select the best mode using the rate distortion optimization (RDO). The RDO mode decision mechanism exhaustively searches the best mode for each 16x16 and 4x4 macroblocks which produces the minimum rate-distortion cost. As a result, the complexity of the Intra-mode decision is extremely high. To reduce the encoding complexity with little RD performance degradation, a machine learning based Intra-mode decision method is proposed which completely bypasses the RDO mechanism. Our results show that the proposed encoder is faster than the RD Optimized H.264 reference encoder with a negligible PSNR loss and a slight increase in bitrate.

The rest of the paper is organized as follows. Section II reviews the principles of operation of Intra-coded macroblocks in I-slices in the H.264/AVC encoding standard. Section III describes the data mining tools and the process of building a decision tree for MB mode estimation. Section IV introduces our MB mode decision algorithm. In section V, we carry out a performance evaluation of the proposed algorithm in terms of its computational complexity and Rate-Distortion results. We compare the performance of our proposed algorithm to the RD-optimized and Hadamard SAD cost methods implemented by the JM 14.2 H.264 reference software. Finally section VI draws our conclusions and outlines our future research plans.

II. PREDICTION OF INTRA CODED MACROBLOCKS

Intra prediction is a pre-processing operation before DCT to change/tweak signal characteristics of input images for compression improvement by tuning them to a DCT basis. The main focus of Intra prediction is to eliminate low frequency components in a predictable way that enables perfect reconstruction of the source picture in the decoder. Generally, a better Intra prediction suppresses low frequency components more. Intra prediction has been present with almost all video compression standards such as MPEG-1, MPEG-2, MPEG-4 Part 2, VC-1 and H.264/AVC. However, actual techniques vary in each standard. In contrast to previous video coding standards (especially H.263 and MPEG-4 Visual), where Intra prediction has been conducted in the transform domain, Intra prediction in H.264/AVC is always conducted in the spatial domain, by referring to neighboring samples of previously decoded blocks that are to the left and/or above the block to be predicted. Since this can result in spatio-temporal error propagation when Inter prediction has been used for neighboring macroblocks, a constrained Intra coding mode can alternatively be selected that allows prediction only from Intra-coded neighboring macroblocks.

There are three types of Intra predictions – Intra 16x16 prediction, Intra 4x4 prediction and Intra 8x8 prediction. In particular Intra 8x8 prediction has been recently introduced for High Profile, the scope of which is outside of our discussion since we are limited to Baseline Profile only. There are nine luma options in 4x4 Intra prediction for each 4x4 luma block while there are four luma options in 16x16 Intra prediction for each 16x16 luma block. Intra prediction algorithm works as follows:

1. Generate a 4x4 predicted block according to a given mode *I*.
2. Calculate sum of absolute differences (SAD_i) between the original 4x4 block and predicted block
3. Compute $COST_{4x4,I} = SAD_i + 4\lambda(QP).R$, where $\lambda(QP)$ is an approximately exponential function of the quantization factor QP, R equals 0 for the most probable mode, and 1 for other modes.
4. Repeat steps 1 to 3 for all the 9 modes and choose the one that has the minimum cost.
5. Generate the predicted MB according to a 16x16 prediction mode.
6. For the residue MB, perform Hadamard transform for each 4x4 block.
7. Extract all the DCs from the 16 4x4 blocks to form another 4x4 block and perform Hadamard transform on it.
8. Repeat steps 5 to 7 for all 4 modes and choose the one with minimum cost as the $COST_{16x16}$.

$$COST_{16x16} \geq \sum_{i}^{16} COST_{4x4,i} + 24\lambda(QP) \qquad (1)$$

To reduce the encoding complexity with little RD performance degradation and bitrate penalty, new Intra mode decision mechanism is proposed in this paper. There are many faster prediction methods have been proposed to successfully reduce the computation for H.264 intra-prediction. Huang *et al.* [9] developed a fast algorithm which performed the context-based adaptive skipping of unlikely prediction modes and simplification of matching operations to save about 50% of the encoding time. Meanwhile, Pan [10] proposed another fast intra-mode decision scheme which using Sobel operator to measure the edge angle of 4x4 blocks and 16x16 macroblock (MB) to reduce the number of probable modes for complexity reduction. Wang [11] proposed a simple edge detection algorithm, which is proposed in MPEG-7 as feature descriptors, to achieve a better result in comparison to the previous algorithm [10]. A more recent study of intra-prediction on fast intra/inter-coding is also published [12], [13]. Tsai *e. al.* [14] proposed a technique based on direction detection algorithm by computing subblock and pixel direction differences.

III. DATA MINING AND MACHINE LEARNING FOR MODE DECISION TREES

While there are no known published results on the use of machine learning in video encoding, there are several techniques on fast intra mode decisions in H.264 encoding [3, 4, 5]. All these approaches reduce the computational cost compared to the H.264 reference software. The complexity reduction, however, is not sufficient to enable the use of complex video encoding features on resource constrained devices. We have developed an innovative approach that is not limited by the selective evaluation approaches.

We propose a machine learning based approach to MB mode decision in H.264/AVC encoding. The complexity of H.264/AVC encoding creates an opportunity for applying machine learning to reduce the complexity. Machine learning can be used to develop decision trees for MB mode decisions without having to evaluate all the possible modes. A decision tree is made by mapping the observations about a set of data in a tree made of arcs, nodes and leaves. The arcs represent the choices that a decision tree can make, nodes represent the classifier and leaves (rectangles) represent the possible modes decided by the decision trees. The tree can have more than one level; in that case, the intermediate nodes represent the decision based on the values of different variables that drives us from the root to the leaf. The tree leaves are the classifications and the branches are the features that lead to a specific classification.

The proposed approach uses C4.5 [4] algorithm for building classifiers. This is one of the commonly used algorithms in data mining. Such systems take as input a collection of cases, each belonging to one of a small number of classes and described by its values for a fixed set of attributes, and output a classifier that can accurately predict the class to which a new case belongs. C4.5 generates classifiers expressed as decision trees. A decision tree is used to classify a case, i.e. to assign a class value to a case depending on the values of the attributes of the case.

The proposed approach was developed based on the insights from our work on MPEG-2 to H.264 transcoding that exploited machine learning tools [13, 14]. The key idea behind this approach is to exploit the correlation between the structural information in a video frame and the corresponding H.264 MB mode decisions and build a classifier or a decision tree. Fig. 2 depicts the process for building the decision trees to be used in the H.264/AVC MB coding mode decisions during encoding process. The decision trees will be used to determine the coding mode of MBs and sub-MBs in I frames of the given H.264/AVC video, based on information gathered during the prior encoding process. This technique determines encoder decisions such as MB coding mode decisions that are computationally expensive by using easily computable features derived from uncompressed video.

The training sets were made using only I-frames of a h.264 coded video sequence. The H.264/AVC coding mode decisions in the training sets were obtained from encoding the video sequence separately for each value of quantization parameter. After extensive experiments, we found that the sequences that contain regions varying from homogenous to high-detail serve as good training sets. Good sample sequences could be Flower and Football. Our goal is to develop a single, generalized, decision tree for each level that can be used for encoding any H.264 video.

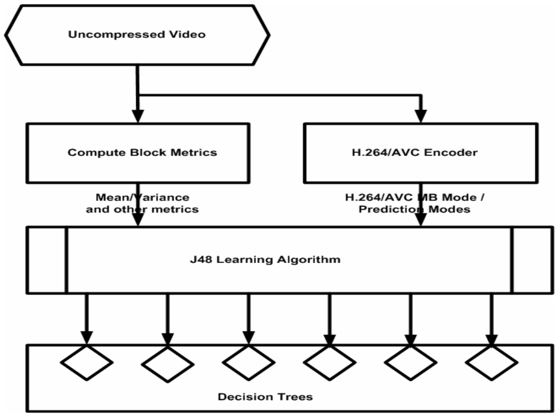


Fig. 2. Process to create decision trees.

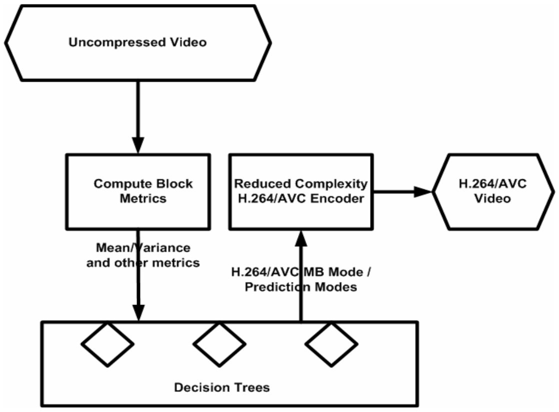


Fig. 3. Using decision trees in complexity reduction mode.

Once a tree is trained, the encoder coding mode decisions that are normally done using cost-based models that evaluate all possible coding options are replaced with a decision tree. Fig. 3 depicts the process of using decision trees in complexity reduction mode. Decision trees are in effect if-else statements in software and require negligible computing resources. We believe this simple approach has the potential to significantly reduce encoding complexity and affect the way encoders are used in mobile devices.

IV. LOW COMPLEXITY MODE DECISION ALGORITHM

This section discusses the proposed low complexity mode prediction algorithm. This goal is achieved by making use of H.264/AVC coding mode along with the means, variances and other attribute values of the pixel data. The energy of a picture and hence of a macroblock is represented by means and variance of the 4x4 sub-blocks of a macroblock. Intra slices in H.264/AVC uses sum of absolute differences (SAD) between the original 4x4 block and predicted block, thus it exploit the spatial correlation of the 4x4 sub-blocks. Therefore, means and variance of 4x4 sub-blocks in a macroblock can be exploited to understand the spatial correlation of two macroblocks or adjacent sub-blocks in a macroblock.

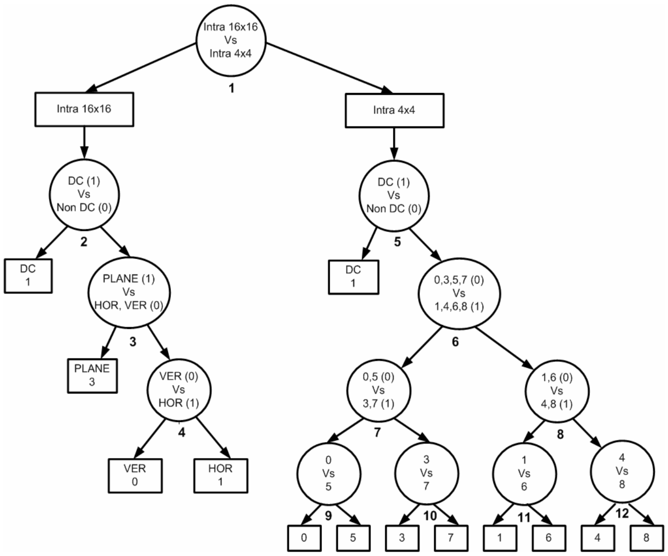


Fig. 4. Decision tree for Intra MB encoding.

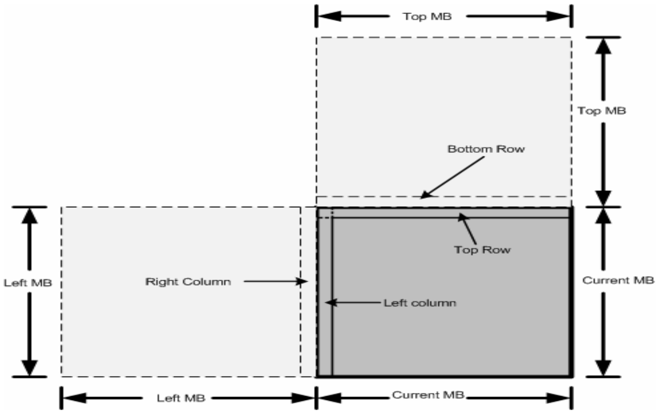


Fig. 5. Attribute selection for Intra 16x16 modes.

Intra MBs in H.264 are coded as Intra 16x16, Intra 4x4, or Intra 8x8. The baseline profile used in mobile devices does not support Intra 8x8 mode and this mode will not be discussed further in this paper. Intra modes also have associated prediction modes; Intra 16x16 has 4 prediction modes and Intra 4x4 has 9 prediction modes. Baseline profile encoders typically evaluate both Intra 16x16 and Intra 4x4 modes and the associated prediction modes before making MB mode decisions. In the proposed machine learning based approach we separate the Intra MB mode and Intra prediction mode decisions. Intra MB mode is determined as Intra 16x16 or Intra 4x4 without evaluating any prediction modes. The appropriate prediction modes for the MB mode are then determined. Since the MB mode is determined first, our approach right away eliminates the computation of any prediction modes for the MB mode that is not selected. If the MB mode is determined to be Intra 16x16, there is no need to evaluate any prediction modes for the 4x4 sub-blocks.

Fig. 4 shows the hierarchical decision tree composed of twelve different trees, used in making H.264/AVC Intra MB mode and prediction mode decisions. Table I shows the decision trees represented by nodes along with the mode decisions. Each node uses its own attributes and hence separate training sets are used for each decision tree.

TABLE I
Decision Trees and Mode Decisions

Node Number	MB Type	Mode Decision
1	Intra	Intra 16x16 Vs Intra 4x4
2	Intra 16x16	DC Vs Non DC
3	Intra 16x16	PLANE Vs HOR, VER
4	Intra 16x16	VER Vs HOR
5	Intra 4x4	DC Vs Non DC
6	Intra 4x4	0,3,5,7 Vs 1,4,6,8
7	Intra 4x4	0,5 Vs 3,7
8	Intra 4x4	1,6 Vs 4,8
9	Intra 4x4	0 Vs 5
10	Intra 4x4	3 Vs 7
11	Intra 4x4	1 Vs 6
12	Intra 4x4	4 Vs 8

The goal of the decision tree is to accelerate the Intra MB mode decisions. This goal is achieved by making use of different attributes of MBs and sub-MBs calculated prior to finalize the mode decisions. The attributes of the MBs and sub-MBs can thus be exploited to understand the spatial correlation of MBs for Intra mode decisions in H.264/AVC. The open source WEKA data mining tool is used to discover a pattern of the attribute values for the H.264/AVC coding mode decisions. Fig. 4 shows the decision tree used in the proposed encoder.

The decision tree consists of twelve WEKA decision trees, shown in Fig. 4 as nodes numbered from 1 to 12. The first WEKA tree (at root node 1) is used to check of the incoming macroblock is going to be handled as 16x16 or as 4x4. If an MB is determined as 16x16, left sub-tree at node 2 is followed, and if necessary up to the node 4 until a mode decision is made. Similarly, if an MB is determined as 4x4 MB, right sub-tree at node 5 is followed up to the node 12, if necessary. As soon as a mode is determined at any stage during tree traversal i.e. leaf is reached, further traversal is aborted and mode number is returned as the coding mode for that MB or sub-MB. The WEKA tool determined the attribute values' thresholds for each of the twelve WEKA trees in the decision tree. Due to space constraints we cannot show all the rules being evaluated in the WEKA decision nodes. The process described in herein should be sufficient for interested people to develop the decision trees and repeat these experiments. The decision tree works as follows:

Attributes: In a digital picture, neighboring blocks have a very high similarity; hence by using these spatial correlations among blocks, most suitable mode can be predicted. The decision tree exploits the spatial similarity of the current candidate macroblock and the prediction pixels that are used to predict different mode decisions. The attributes chosen as input to WEKA tool are based on the pixel intensity of the current MB, sub-MB, rows (top, bottom) and columns (right, left) of the current as well as of the adjacent MBs and sub-MBs.

Fig. 5 shows the schematic diagram for the calculation of attributes in 16x16 macroblocks. The dark rectangle shows the macroblock being evaluated while light colored rectangles are the top and left macroblocks adjacent to the current candidate macroblock. We calculate the 16 means and 16 variances in the current macroblock of 4x4 sub-blocks. We also provide the difference between the bottom row of top MB and top row of the current MB. Similarly, we calculate the difference between the left column of current MB and the right column of left MB.

These two differential metrics give a strong hint about the horizontal or vertical prediction of the current MB.

The decision trees obtained by using various attributes are discussed below. All of the decision trees are implemented as binary trees. Tables II – VII show the performance of the training set (from node 1 to node 6) in terms of CCI % (Correctly Classified Instances).

TABLE II
Node 1 Decision Tree Statistics

Node	Sequence	QP	Number of Samples	Number of Leaves	CCI (%)
1	Flower	20	3168	16	97.601
		24	3168	11	97.9167
		28	3168	9	97.5379
		32	3168	7	97.5694
		36	3168	5	95.4545
		40	3168	8	92.7715

TABLE III
Node 2 Decision Tree Statistics

Node	Sequence	QP	Number of Samples	Number of Leaves	CCI (%)
2	Flower	20	1182	10	87.1404
		24	1268	9	85.1735
		28	1385	12	83.1769
		32	1534	7	83.442
		36	1772	8	84.8194
		40	2096	10	84.9237

TABLE IV
Node 3 Decision Tree Statistics

Node	Sequence	QP	Number of Samples	Number of Leaves	CCI (%)
3	Flower	20	983	5	88.6063
		24	1044	6	89.1762
		28	1137	6	86.8953
		32	1223	9	90.5151
		36	1108	6	78.7004
		40	1700	6	91.5882

Node 1: The inputs for this node are the variance of 16 4x4 sub-Mbs variances and mean of 16 4x4 sub-MBs means. Since 16x16 mode is selected for a homogenous region, therefore, these two attributes proved to be very helpful in determining the intensity variance for a macroblock. For a 16x16 MB, variance of the pixel values tends to be very low, which is also a very helpful indicator in determining the macroblock mode. The output of this node determines the type of the macroblock mode type i.e. 16x16 or 4x4. Table II shows the statistics of the decision tree for node 1.

Node 2: This node differentiates between the DC mode and non-DC modes for 16x16 macroblock. Since DC mode has no specific direction, it cannot be predicted with a spatial correlation. The inputs for this node are the variance of 16 4x4 sub-Mbs variances and mean of 16 4x4 sub-MBs means. There are two additional attributes which are calculated by subtracting the mean values of the bottom row of the current MB from the bottom row of top MB. Similarly, we also calculate the difference between the mean values of the right column of left MB and right column of current MB. Table III shows the statistics of the decision tree for node 2.

Node 3: The output of this node classifies Plane mode as one class and Horizontal, Vertical modes as another class by using the same inputs as that of node 2. The statistics of the classifier of node 3 is shown in table IV.

TABLE V
Node 4 Decision Tree Statistics

Node	Sequence	QP	Number of Samples	Number of Leaves	CCI (%)
4	Flower	20	1750	12	75.9429
		24	1895	15	73.6148
		28	2035	15	72.3833
		32	2269	10	74.5703
		36	2647	11	73.4794
		40	3082	12	74.0104

Node 4: This node also uses the same attributes as inputs as that of node 2 and node 3. The classifier at this node differentiates between horizontal and vertical prediction modes. The output of this node is either mode 0 (vertical) or mode 1 (horizontal) for 16x16 macroblock. The statistics of the classifier of node 3 is shown in table V.

TABLE VI
Node 5 Decision Tree Statistics

Node	Sequence	QP	Number of Samples	Number of Leaves	CCI (%)
5	Flower	20	4480	5	80.6027
		24	4304	8	80.948
		28	4048	9	80.5089
		32	3856	5	78.7863
		36	3488	4	77.9243
		40	2976	6	75.5376

TABLE VII
Node 6 Decision Tree Statistics

Node	Sequence	QP	Number of Samples	Number of Leaves	CCI (%)
6	Flower	20	1412	6	65.5807
		24	1403	7	66.6429
		28	1338	9	65.0972
		32	1217	7	63.3525
		36	993	6	65.861
		40	760	4	64.7368

Node 5: This node differentiates between DC and non-DC mode for a 4x4 sub-block in a macroblock. The inputs for this node are the mean and variance of the current sub-block as well as a third attribute calculated by taking into account the intensity values of rows and columns in angular directions. The statistics for this node are shown in table VI.

Node 6: This node classifies non-DC modes for Intra 4x4 mode into two classes. We divided the Intra 4x4 non-DC modes such that the 4 adjacent modes from Fig. 1 (b) go to each class. Therefore, we classified 3, 7, 0, 5 as one class while 4, 6, 1, 8 as another class. The statistics for this classifier is shown in table VII.

Since the MB mode decisions depend upon the quantization parameter (QP) used in H.264/AVC encoding, the mean and variance threshold will have to be different at each QP. Two solutions are possible: 1) develop a single decision tree and adjust the mean, variance and other attributes' threshold used by the trees based on the QP and 2) develop the decision trees for each QP and use the appropriate decision tree depending upon the QP selected. For the first option, to use a single decision tree, the decision tree has to be developed for a mid QP value of 25 and then have to be adjusted for other values of QP. Since the quantization step size in H.264/AVC doubles when QP is increased by 6, thresholds are adjusted by 2.5% for a change in QP of 1. For QP values higher than 25, the thresholds are decreased and for the QP values lower than 25 thresholds are proportionally increased. For the second option, the decision trees are built for 6 different values of QP i.e. 20, 24, 28, 32, 36 and 40. Since the standard conformance procedure for H.264/AVC standard mostly uses these values of QP, we decided to use the same procedure. The different trees were implemented in reference software JM 14.2 [6] by using the function pointers of C. After extensive experimentation to evaluate both of the methods mentioned above, we came to conclude that the using multiple decision trees for separate QP values produces better results.

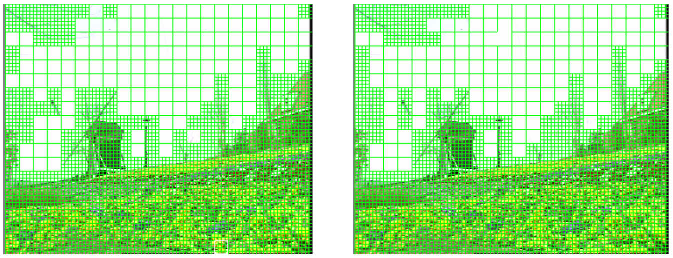

(a) flower.cif (original QP=28) (b) flower.cif (proposed QP = 28)

Fig. 7. Macroblock partitions generated by the proposed algorithm for the I-frame in flower sequence for Intra 16x16 vs. Intra 4x4 mode decision.

Fig. 7 shows an example of the results obtained by applying our proposed algorithm. Fig. 7a illustrates the Intra mode selection (16x16 vs 4x4) in the original flower sequence of CIF format encoded by using the JM reference software. Fig. 7b shows the Intra mode selection by using our proposed algorithm. From this comparison, it is clear that our algorithm generates very similar results to those obtained by using the JM reference software.

V. PERFORMANCE EVALUATION

The proposed low complexity MB coding mode decision algorithm is implemented in the H.264/AVC reference software, version JM 14.2 [6]. Fig. 2 and Fig. 3 show the overall operation of the proposed encoder. The H.264/AVC video is decoded and the information required by the decision trees is gathered in this stage as shown in Fig. 2. The additional computation here is the computation of the mean and variance of the 4x4 sub-blocks of the residual MBs. The MB coding mode decision determined by the decision trees is used in the low complexity H.264 encoding stage as shown in Fig. 3. This is an H.264 reference encoder with the MB mode decision replaced by simple mode assignment from the decision tree. The H.264 video encoder takes as input the decoder H.264/AVC video (pixel data) and the MB mode decision from the decision tree and encodes the H.264 video. The H.264/AVC Intra prediction mechanism which uses the SAD (Sum of Absolute Differences) calculation is not used and the encoder performs the Intra prediction just for the final MB mode determined by the decision tree.

The performance of the proposed very low complexity encoder is compared with a reference H.264/AVC encoder. We compare the performance of our proposal to the H.264/AVC encoder with the RD Optimization enabled. The metrics used to evaluate the performance are the reduction in the computational cost and rate distortion function which are measured by PSNR and bit rate (BR). The time results reported are for the H.264 encoding for both the proposed and reference encoders.

We have conducted an extensive set of experiments with videos representing wide range of motion, texture, and color. Experiments were conducted to evaluate the performance of the proposed algorithm when transcoding videos at commonly used resolutions: CCIR-601, CIF, and QCIF. The input to the encoder is a YUV sequence. Since the proposed encoder addresses the problem of complexity reduction for Intra prediction, all the frames are encoded using I frames coding. The experiments have shown that the proposed approach performs extremely well to reduce the complexity across all QP values and resolution formats.

The sequences have been encoded with H.264 using the QP factors ranging from 20 up to 40 in steps of 4. This corresponds to the H.264 QP range used in most practical applications. All frames in sequences were encoded as I-frames by specifying the value of IntraPeriod to 1. The rate control was disabled for all the simulations. The ProfileIDC was set to Baseline for all the simulations. The simulations were run on Intel Core2 Duo machine running with 2.40 GHz processor with 4 GB RAM. The results are reported for six different sequences: two for each of the three resolutions mentioned above. The configuration parameters for the proposed and reference encoder are the same. Due to the space constraints, RD curves are shown for 3 CIF and QCIF sequences (Akiyo, Mobile, Stefan) and for 2CCIR (Mobile, Flower) sequences. However, average speed up, BR increase and PSNR loss are shown for all sequences, we used for the simulations, in the tables below.

The comparison metrics were produced and tabulated based on the difference of percentage coding time (ΔTime), the PSNR difference (ΔPSNR) and the percentage bit-rate difference (ΔBR). PSNR and bit-rate differences are calculated according to the numerical averages between the RD curves derived from reference JM encoder, the algorithm under study (modified JM encoder with RD Optimization disabled). The detail procedures in calculating these differences can be found in the JVT documents by Bjontegaard [7], which is recommended by JVT Test Model Ad Hoc Group [8]. Note that PSNR and bit-rate differences should be regarded as equivalent, i.e., there is either a decrease in PSNR or an increase in bit-rate, but not both at the same time. To evaluate the performance of our proposed, we divide our classifiers into following four categories:

A. Top Level Classifiers

The classifiers at this level are the decision trees that decide MB mode partition type i.e. Intra 16x16 or Intra 4x4. Rest of the mode decisions are determined by the default reference implementation in JM 14.2.

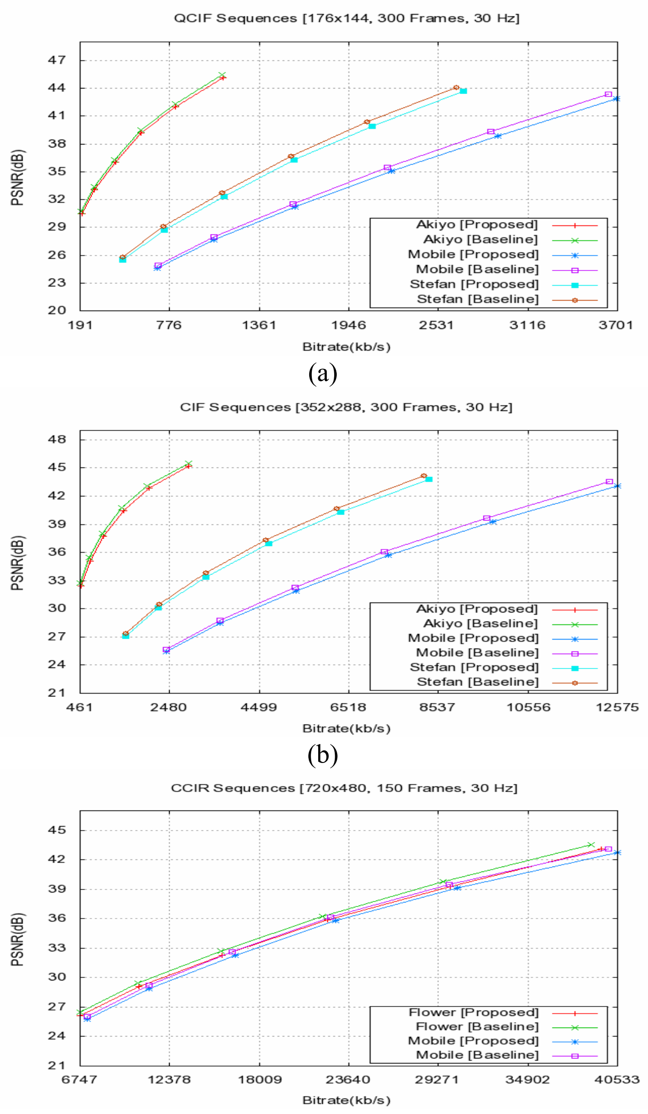
Fig. 8 shows the RD performance results for the reference and proposed encoder for selected sequences. As seen from the Fig. 8, PSNR obtained with the proposed encoder deviates slightly from the results obtained when applying the considerable more complex reference encoder. Table VIII summarizes the average statistics for the top level classifier. Compared with the reference encoder, the proposed encoder has average PSNR drop 0.32dB with the penalty of little bit more than 1% of average BR and average speed up of about 68% for QCIF sequences. Similarly, the average results for CIF and CCIR-601 sequences are shown in the same table. The negligible drop in PSNR is more than offset by the reduction in computational complexity. As shown in the table VIII, the encoding time is reduced by more than 60% with RD optimization.

B. Intra 16x16 Classifier

This classifier at this level enables the encoder to classify Intra 16x16 decisions based on the proposed approach. This classifier also includes the decisions made by the top level classifier while Intra 4x4 mode decisions are made by the reference encoder. RD performance results and statistics for this level of classifier are very similar to that of top level classifier. For the sake of brevity in this paper, we are omitting the results for this classifier. The encoding time is reduced by more than 60% with RD optimization.

C. Intra 4x4 Classifier

This classifier at this level enables the encoder to classify Intra 4x4 decisions only based on the proposed approach. This classifier also includes the decisions made by the top level classifier while Intra 16x16 mode decisions are made by the reference encoder. Fig. 9 shows the RD performance results for the reference encoder and proposed encoder for selected sequences. Table IX summarizes the average statistics for the top level classifier. Compared with the reference encoder, the proposed encoder has average PSNR drop 0.24dB with the penalty of about 29% of average BR and average speed up of about 65% for QCIF sequences. Similarly, the average results for CIF and CCIR-601 sequences are shown in the same table. The drop in PSNR is negligible; however massive bit-rate increase does not make this classifier viable for proposed encoder although the encoding time is reduced by more than 65% for all sequences.



(a)

(b)

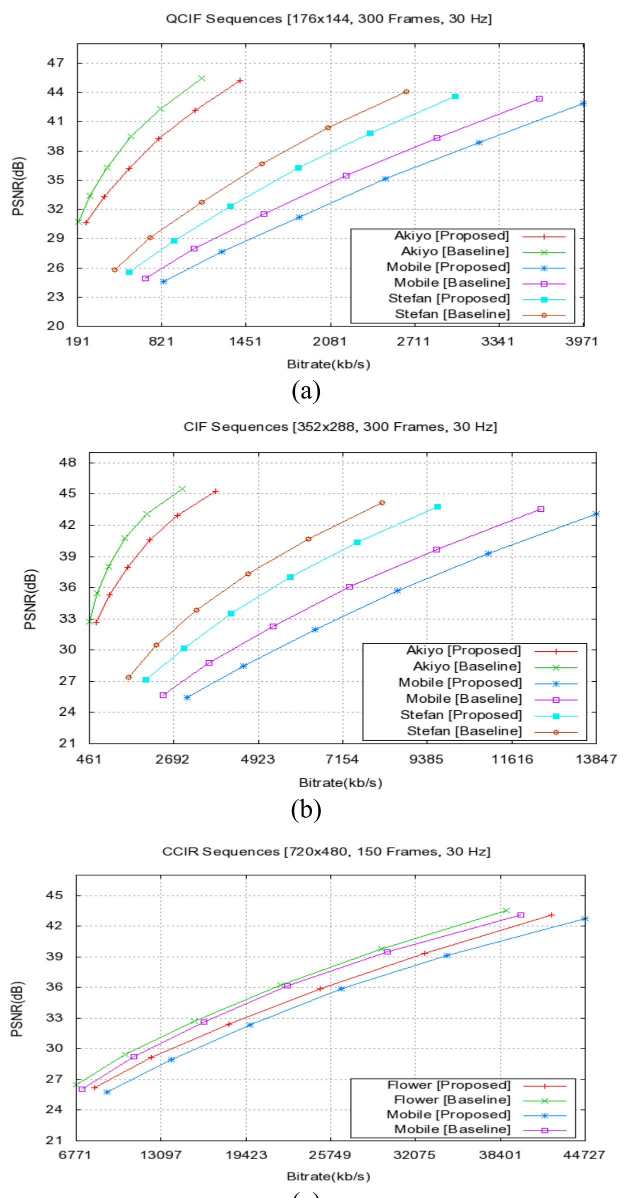
(c)

Fig. 8. RD performance for Top Level Classifier

D. Combined Classifier

At this level, all Intra mode decisions are determined by the proposed classifiers i.e., first of all top level classifier determines whether the current MB is going to be Intra 16x16 or Intra 4x4, then according to the selected mode, next level classifier decides about the prediction directions. RD performance results and statistics for this level of classifier are very similar to that of Intra 4x4 classifier. For the sake of brevity in this paper, we are omitting the results for this classifier.

One important observation in above mentioned four different classifiers is the effect of bit-rate increase as a result of complexity reduction in all classifiers. While for top level and Intra 16x16 classifiers, bit-rate penalty is within reasonable limits but for Intra 4x4 and combined classifiers, bit-rate penalty is relatively much larger than the other two classifiers. In summary, the proposed algorithm is about two times faster than JM14.2 reference software while the increase in BR is nearly negligible in the case of top level and Intra 16x16 classifiers. In the case of Intra 16x16 and combined classifiers, we get the same amount of complexity reduction with tangible increase in the bit-rate. Comparing our top level and Intra 16x16 classifiers with the proposed algorithm for Pixel-Based Direction Detection (PDD) in [14], the computation time of the proposed methods is also reduced albeit by very minor margin. However, our proposed classifiers are more cost effective in terms of bit-rate penalty by the factor of 2 as shown in Table X. This comparison is based only on QCIF and CIF sequences.



(a)

(b)

(c)

Fig. 9. RD performance for Intra 4x4 Classifier

VI. CONCLUSION

In this paper, we proposed a novel macroblock partition mode decision algorithm for Intra prediction in H.264/AVC. The proposed algorithm used data mining techniques to exploit the correlation between the H.264/AVC MB statistics and H.264 MB coding modes. The WEKA tool was used to develop decision tree for H.264/AVC coding mode decision. The proposed algorithm has very low complexity as it only requires the calculation of MB statistics such as means, variances and differences of border pixels. The proposed encoder was evaluated using YUV sequences at QCIF, CIF and CCIR resolutions. Our results show that the proposed algorithm is able to maintain a good picture quality while considerably reducing the computational complexity by 65% on average. The reduction in computational cost has negligible impact on the quality and bit-rate of the encoded video. Our results show that the proposed algorithm maintains its performance across all resolutions. The proposed approach is novel and the basic idea can also be used to refine other techniques in video encoding.

Our future plans will focus on further reducing the complexity of the proposed encoder by refining the machine learning algorithms and employing further statistical analysis techniques for motion estimation in Inter prediction.

REFERENCES

[1] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, Version 11.

[2] Thomas Wiegand, Gary Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard”, *IEEE Transactions on Circuits Systems and Video Technology*, vol. 13, pp. 560-576, July 2003.

[3] Ian H. Witten and Eibe Frank, “Data Mining: Practical Machine Learning Tools and Techniques”, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[4] Quinlan J.R., “C4.5: Programs for Machine Learning”, *Morgan Kaufman Publishers Inc.*, San Francisco, CA, 1993.

[5] Kalva, H, Lee J.B, “The VC-1 and H.264 Video Compression Standards for Broadband Video Services”, Springer, 2008.

[6] Joint Model (JM) – H.264/AVC Reference Software, http://iphome.hhi.de/suehring/tml/download

[7] Gisle Bjontegaard, “Calculation of Average PSNR Differences between RD Curves”, ITU-T SC16/Q6, 13th VCEG Meeting, Austin, Texas, USA, April 2001, Doc. VCEG-M33.

[8] JVT Test Model Ad Hoc Group, “Evaluation Sheet for Motion Estimation”, Draft version 4, 2003.

[9] Y. W. Huang, B. Y. Hsieh, T. C. Chen, and L. G. Chen, “Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra-frame coder,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 378–401, Mar. 2005.

[10] 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/AVCvideo coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 813–822, Jul. 2005.

[11] J. C. Wang, J. F. Wang, J. F. Yang, and J. T. Chen, “A fast mode decision algorithm and its VLSI design for H.264/AVC intra-prediction,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1414–1422, Oct. 2007.

[12] Choi, J. Lee, and B. Jeon, “Fast coding mode selection with rate-distortion optimization for MPEG-4 part-10 AVC/H.264,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1557–1561, Dec. 2006.

[13] G. Fernandez-Escribano, H. Kalva, P. Cuenca, and L. Orozco-Barbosa, “Very Low Complexity MPEG-2 to H.264 Transcoding Using Machine Learning”, *Proceedings of the ACM Multimedia 2007*, Santa Barbara, CA, October 2006, pp. 931-940.

[14] G. Fernandez-Escribano, C. Holder, J. L. Martinez, R. Jillani, Hari Kalva, P. Cuenca, “Video Encoding and Transcoding Using Machine Learning”, In Proceedings of the 9th Intl. Workshop on Data Mining 2008, August 2008.

TABLE VIII
COMPARISON RESULTS FOR TOP LEVEL CLASSIFIER

Sequence	QCIF			CIF			CCIR		
	ΔPSNR dB	ΔBR %	ΔTIME %	ΔPSNR dB	ΔBR %	ΔTIME %	ΔPSNR dB	ΔBR %	ΔTIME %
Akiyo	-0.292	1.38	-60.5	-0.310	3.08	-60.91			
Coastguard	-0.271	0.565	-64.48	-0.277	0.518	-65.52			
Container	-0.280	1.772	-62.537	-0.267	1.542	-64.143			
Flower	-0.465	0.865	-67.167	-0.474	1.013	-67.408	-0.424	0.846	-67.866
Foreman	-0.223	0.858	-62.256	-0.260	1.284	-63.277			
Hall Monitor	-0.263	2.215	-62.333	-0.244	2.624	-62.964			
Mobile	-0.404	0.754	-67.429	-0.377	0.766	-68.050	-0.355	0.911	-67.835
Mother Daughter	-0.286	0.812	-61.269	-0.265	2.716	-61.026			
Silent	-0.303	0.198	-62.761	-0.314	0.884	-63.427			
Stefan	-0.430	1.001	-65.914	-0.402	0.726	-66.052			
Table	-0.289	1.232	-62.533	-0.263	1.138	-64.428			
Average	**-0.319**	**1.06**	**-63.56**	**-0.31**	**1.48**	**-64.29**	**-0.39**	**0.88**	**-67.85**

TABLE IX
COMPARISON RESULTS FOR INTRA 4X4 CLASSIFIER

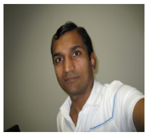
Sequence	QCIF			CIF			CCIR		
	ΔPSNR dB	ΔBR %	ΔTIME %	ΔPSNR dB	ΔBR %	ΔTIME %	ΔPSNR dB	ΔBR %	ΔTIME %
Akiyo	-0.130	33.640	-62.208	-0.143	41.829	-62.066			
Coastguard	-0.127	31.531	-66.120	-0.101	35.433	-67.104			
Container	-0.221	30.781	-64.024	-0.172	32.450	-64.872			
Flower	-0.402	13.413	-69.142	-0.412	13.497	-68.486	-0.363	14.610	-69.615
Foreman	-0.240	48.931	-64.109	-0.199	41.902	-64.440			
Hall Monitor	-0.118	39.265	-64.103	-0.049	48.467	-63.896			
Mobile	-0.375	14.720	-69.383	-0.349	19.223	-69.651	-0.330	19.432	-69.523
Mother Daughter	-0.168	33.426	-62.947	-0.055	51.746	-61.766			
Silent	-0.205	32.184	-64.814	-0.183	33.188	-64.962			
Stefan	-0.405	18.798	-67.696	-0.321	25.540	-67.398			
Table	-0.241	21.082	-63.972	-0.169	24.343	-65.270			
Average	**-0.24**	**28.89**	**-65.32**	**-0.20**	**33.42**	**-65.45**	**-0.35**	**17.02**	**-69.57**

TABLE X
Comparison Results with Other Algorithm [14]

Sequence	[14] PDD			Top Level Classifier			Intra 16x16 Classifier		
	ΔPSNR dB	ΔBR %	ΔTIME %	ΔPSNR dB	ΔBR %	ΔTIME %	ΔPSNR dB	ΔBR %	ΔTIME %
QCIF	-0.21	3.18	-62.67	-0.32	1.06	-63.56	-0.33	1.82	-64.02
CIF	-0.26	3.56	-63.35	-0.31	1.48	-64.29	-0.33	2.68	-64.97

[15] C. Kim and C.-C. J. Kuo, “Feature-Based intra-/intercoding mode selection for H.264/AVC,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 441–453, Apr. 2007.

[16] An-Chao Tsai, Jhing-Fa Wang, Jar-Ferr Yang, and Wei Guang Lin, “Effective Subblock-Based and Pixel-Based Fast Direction Detections for H.264 Intra Prediction”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 975–982, Jul. 2008.



Rashad M Jillani received his MSc degree in Computer Science from IIU, Islamabad, Pakistan in 2000. He is currently a PhD candidate student in computer engineering at the Department of Computer Science & Engineering, Florida Atlantic University.

His research interests include video compression, video transcoding and complexity reduction in mobile video devices.

Hari Kalva (SM’92-M’00-SM’05) is an Associate Professor in the Department of Computer Science and Engineering at Florida Atlantic University. Dr. Kalva is an expert on digital audio-visual communications systems with over 15 years of experience in multimedia research, development, and standardization. He has made key contributions to the MPEG-4 Systems standard and also contributed to the DAVIC standards development. His research interests include pervasive media delivery, content adaptation, video transcoding, video compression, and communication.

Dr. Kalva received a Ph.D. and an M.Phil. in Electrical Engineering from Columbia University in 2000 and 1999, respectively. He received an M.S. in Computer Engineering from Florida Atlantic University in 1994, and a B. Tech. in Electronics and Communications Engineering from N.B.K.R. Institute of Science and Technology, S.V. University, Tirupati, India in 1991.