A Fast MB Mode Decision Algorithm for MPEG-2 to H.264 P-Frame Transcoding

Gerardo Fernández-Escribano, Hari Kalva, Pedro Cuenca, *Member, IEEE*, Luis Orozco-Barbosa, *Member, IEEE*, and Antonio Garrido

***Abstract*—The H.264 standard achieves much higher coding efficiency than the MPEG-2 standard, due to its improved inter-and intra-prediction modes at the expense of higher computational complexity. Transcoding MPEG-2 video to H.264 is important to enable gradual migration to H.264. However, given the significant differences between the MPEG-2 and the H.264 coding algorithms, transcoding is a much more complex task and new approaches to transcoding are necessary. The main problems that need to be addressed in the design of an efficient heterogeneous MPEG-2/H.264 transcoder are: the inter-frame prediction, the transform coding and the intra-frame prediction. In this paper, we focus our attention on the inter-frame prediction, the most computationally intensive task involved in the transcoding process. This paper presents a novel macroblock (MB) mode decision algorithm for P-frame prediction based on machine learning techniques to be used as part of a very low complexity MPEG-2 to H.264 video transcoder. Since coding mode decisions take up the most resources in video transcoding, a fast MB mode estimation would lead to reduced complexity. The proposed approach is based on the hypothesis that MB coding mode decisions in H.264 video have a correlation with the distribution of the motion compensated residual in MPEG-2 video. We use machine learning tools to exploit the correlation and construct decision trees to classify the incoming MPEG-2 MBs into one of the several coding modes in H.264. The proposed approach reduces the H.264 MB mode computation process into a decision tree lookup with very low complexity. Experimental results show that the proposed approach reduces the MB mode selection complexity by as much as 95% while maintaining the coding efficiency. Finally, we conduct a comparative study with some of the most prominent fast inter-prediction methods for H.264 presented in the literature. Our results show that the proposed approach achieves the best results for video transcoding applications.**

***Index Terms*—H.264, inter-frame, machine learning, MPEG-2, transcoding.**

I. Introduction

THE MPEG-2 video coding standard is widely used in digital video applications including digital TV, DVD, and HDTV applications. While MPEG-2 is widely deployed, a new video coding standard known as H.264 or MPEG-4 AVC is gaining significant interest due to its improved performance [1]. The H.264 video provides equivalent video quality at 1/3 to 1/2 of MPEG-2 bit rates. However, these gains come with a significant increase in encoding and decoding complexity [2].

In the near future, given the wide deployment of MPEG-2 infrastructure, MPEG-2 and H.264 are likely to coexist even as H.264 systems are deployed. The coexistence of these technologies necessitates systems that can leverage MPEG-2 and H.264 to provide innovative digital video services making use of MPEG-2 for current devices and H.264 for new generation receivers. The key to making this seamless is by transcoding MPEG-2 to H.264 at the appropriate points in the video distribution infrastructure. Depending on the application needs, the video can be transcoded at the sender, in the network, or even at the receiver. However, given the significant differences between both encoding algorithms, the transcoding process of such systems is a much more complex task than other heterogeneous video transcoding processes [3]–[6].

The main elements that require to be addressed in the design of an efficient heterogeneous MPEG-2 to H.264 transcoder are [7]: the inter-frame prediction, the transform coding and the intra-frame prediction. Each one of these elements requires to be examined and various research efforts are underway. In this paper, we focus our attention on a part of the inter-frame prediction: the *macroblock (MB) mode decision in P-frames*, one of the most stringent tasks involved in the transcoding process.

A video transcoder is comprised of a decoding stage followed by an encoding stage. The decoding stage of a transcoder can perform full decoding to the pixel level or partial decoding to the coefficient level. Partial decoding is used in compressed domain transcoding where the transform coefficients in the input format are directly transcoded to the output format. This transformation is straightforward when the input and output formats of the transcoder use the same transform (e.g., MPEG-2 to MPEG-4 transcoding) [5]. When these transforms differ substantially, the compressed domain transcoding becomes computationally expensive. The utility of this compressed domain transcoding is limited to intra-MB transcoding. For predicted MBs, the transcoding in compressed domain becomes prohibitively expensive. The substantial differences in MPEG-2 and H.264 make even intra-transcoding in the compressed domain relatively expensive [8]; pixel domain transcoding is shown to produce better results [9]. Pixel domain transcoders have a full decoding stage followed by a reduced complexity encoding stage. The complexity reduction is achieved by reusing the information gathered from the decoding stage. It is assumed that the input video is encoded with reasonable R-D optimization.

Manuscript received December 15, 2006; revised April 30, 2007. This work was supported in part by the Spanish MEC and European Commission FEDER under Grants Consolider Ingenio-2010 CSD2006-00046 and TIN2006-15516-C04-02, and in part by JCCM under Grant PAI06-0106. This paper was recommended by Associate Editor H. Sun.

G. Fernández-Escribano, P. Cuenca, L. Orozco-Barbosa, and A. Garrido are with the Instituto de Investigación en Informática de Albacete, Universidad de Castilla-La Mancha, 02071 Albacete, Spain (e-mail: gerardo@dsi.uclm.es; pcuenca@dsi.uclm.es; lorozco@dsi.uclm.es; antonio@dsi.uclm.es).

H. Kalva is with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: hari@cse.fau.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2008.918115

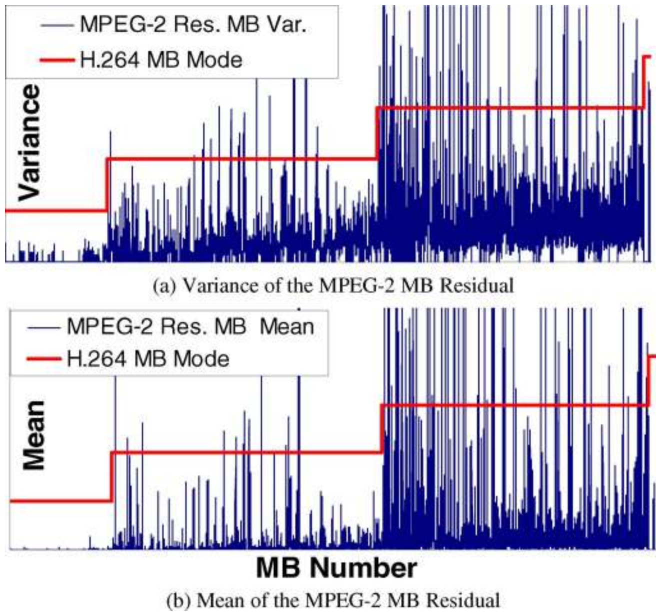


Fig. 1. Relationship between MPEG-2 MB residual and H.264 MB coding mode.

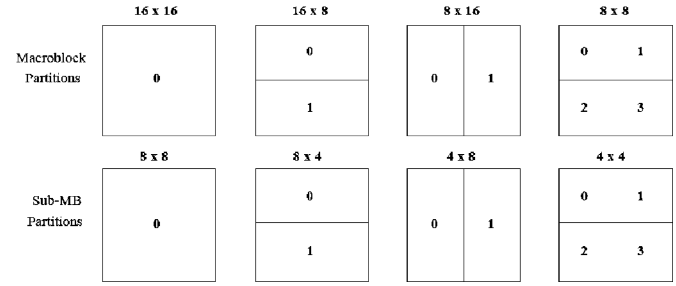


Fig. 2. MB partitions, sub-MB partitions and partition scans for INTER modes.

The MPEG-2 to H.264 complexity reduction techniques reported in the literature fall into two categories: 1) MB mode mapping in H.264 based on the MB modes of the incoming video [10] and 2) Selective evaluation of MB modes in H.264 based on heuristics [11]. Because of the large number of inter-and intra-MB coding modes supported by H.264, there is no one-to-one mapping between MPEG-2 and H.264 MB modes. A direct mapping leads to either a suboptimal decision if the mapped mode is the final MB mode or an increase in complexity if additional evaluations have to be made to improve the mode decision. Selective evaluation is based on the observation that certain MB modes are less likely to occur for a class of videos and bit rates. If the selective evaluation is aggressive in limiting the number of allowed modes, the performance is suboptimal. On the contrary, increasing the number of allowed modes increases the complexity.

In this work, we present an innovative approach that is not limited by the inefficiencies of mode mapping or selective evaluation approaches. The proposed approach is based on the hypothesis that MB coding mode decisions in H.264 video have a correlation with the distribution of the motion compensated residual in MPEG-2 video. Exploiting this correlation together with the MB coding modes of MPEG-2 could lead to a very low complexity transcoder.

Fig. 1 shows a plot of the mean and variance of the MPEG-2 MB residual in the input video and the H.264 MB coding mode of the corresponding MB in the transcoded video. The X axis shows the index of the MB in the set of MBs ordered according to their MB mode. As the coding mode changes, the shift in the mean and variance of the corresponding MB can be clearly seen. This correlation can be effectively exploited using machine learning approaches. Thus, the H.264 MB mode computation problem is posed as a data classification problem where the MPEG-2 MB coding mode and residual have to be classified into one of the several H.264 coding modes. The proposed transcoder is developed based on these principles and reduces the H.264 MB mode computation process into a decision tree lookup with very low complexity.

The rest of the paper is organized as follows. Section II reviews the principles of operation of the interframe prediction in the H.264 video coding standard. In Section III, we review some of the most relevant proposals aiming to speed-up the inter-frame prediction process in H.264. Section IV introduces our MB mode decision algorithm for inter-frame prediction based on machine learning techniques, specifically designed for MPEG-2 to H.264 transcoders. In Section V, we carry out a performance evaluation of the proposed algorithm in terms of its computational complexity and rate-distortion (R-D) results. We compare the performance of our proposal to the reference transcoder with the encoding stage using the H.264 reference implementation. We also analyze a comparative study with some of the most prominent fast inter-prediction methods for H.264 presented in the literature. Finally, Section VI draws our conclusions and outlines our future research plans.

II. Overview of Interframe Prediction in H.264

In the H.264 standard, the MB mode decision in inter-frames is the most computationally expensive process due to the use of the variable block-size motion estimation, quarter-pixel resolution motion vectors (MVs), intra-prediction, etc.

H.264 uses block-based motion compensation, the same principle adopted by every major coding standard since H.261. Important differences from earlier standards include the support for a range of block sizes (down to 4×4) and the use of multiple reference frames. H.264 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 MB (16×16 samples) may be split up in 4 ways: 16×16, 16×8, 8×16 or 8×8. Each of the subdivided regions is a *MB partition*. If the 8×8 mode is chosen, each of the four 8×8 MB partitions within the MB may be further split in 4 ways: 8×8, 8×4, 4×8 or 4×4 (known as *sub-MB partitions*). These partitions and subpartitions give rise to a large number of possible combinations within each MB (see Fig. 2). This method of partitioning MBs into motion compensated subblocks of varying size is known as *tree structured motion compensation*.

A separate MV (previously calculated in the motion estimation module) is required for each partition or subpartition. Each MV must be coded and transmitted; in addition, the choice of partition(s) must be encoded in the compressed bit stream. Choosing a large partition size (e.g., 16×16, 16×8, 8×16)

means that a small number of bits are required to signal the choice of MV(s) and the type of partition; however, the motion compensated residual may contain a significant amount of energy in areas with high detail. 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 MVs and choice of partition(s). The partition size used therefore has a significant impact on the compression performance. In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for areas with high detail.

If the video source sampling is 4:2:0, the resolution of each chroma component in a MB (Cr and Cb) is half that of the luminance (luma) component. Each chroma block is partitioned in the same way as the luma component, except that the partition sizes have exactly half the horizontal and vertical resolution (an 8×16 partition in luma corresponds to a 4×8 partition in chroma; an 8×4 partition in luma corresponds to 4×2 in chroma; and so on). The horizontal and vertical components of each MV (one per partition) are halved when applied to the chroma blocks.

Each partition in an inter-coded MB is predicted from an area of the same size in a reference picture. The offset between the two areas (the MV) has 1/4-pixel resolution (for the luma component). If the video source sampling is 4:2:0, 1/8 pixel samples are required in the chroma components (corresponding to 1/4-pixel samples in the luma). The luma and chroma samples at subpixel positions do not exist in the reference picture and so it is necessary to create them using interpolation from nearby image samples. Subpixel motion compensation can provide significantly better compression performance than integer-pixel compensation but at the expense of increased complexity. Quarter-pixel accuracy outperforms half-pixel accuracy.

Encoding a MV for each partition can take a significant number of bits, especially if small partition sizes are chosen. Motion vectors for neighboring partitions are often highly correlated and so each MV is predicted from vectors of nearby, previously coded partitions. The method of forming the MV prediction (MVp) depends on the motion compensation partition size and on the availability of nearby vectors.

In addition, H.264 also allows INTRA modes in inter-frames. Therefore, although the current MB belongs to an inter-slice, H.264 must examine all intra-prediction modes: an MB may make use of 4×4 and 16×16 block prediction modes, referred to as *Intra_4* $\times$ *4* and *Intra_16* $\times$ *16*, respectively. Recently, the *Intra_8* $\times$ *8* block prediction mode has been added as part of the Fidelity Range Extension (FRExt) of the standard. There are nine 4×4 and 8×8 possible block prediction directions and four 16×16 block prediction directions. These intra-prediction modes include a directional prediction greatly improving the prediction in the presence of directional structures. Finally, H.264 also allows a SKIP mode in inter-frames referring to the 16×16 mode where no motion and residual information is encoded.

The basic process for inter-frame prediction in H.264 can be briefly summarized as follows: the encoder encodes an MB using all possible modes (inter, intra-and skip), such as variable block size motion estimation, variable block size intra-prediction, and multiple reference frames. The MB coding mode which produces the least ***cost*** will be used in the final coding. In the H.264 JM reference software (version 10.2) [12], two methods have been defined to evaluate the cost for MB mode decision: R-D-cost and SAE-cost. In the following, we describe these two methods.

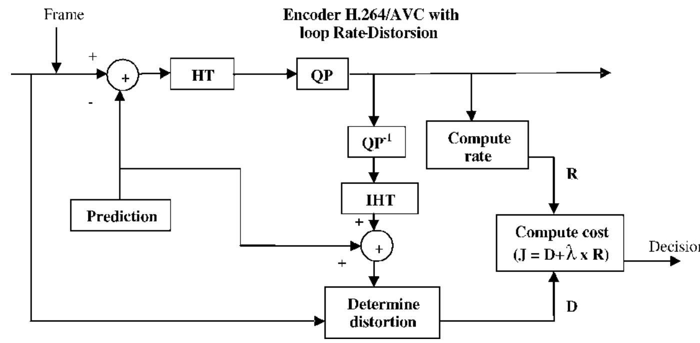


Fig. 3. R-D-cost method in the H.264 JM reference software.

A. R-D-Cost

The R-D optimization method is based on a Lagrange multiplier [13], [14]. The H.264 standard can make use of this optimization method to choose the best MB mode decision. In this way, the H.264 standard selects the MB mode exhibiting the minimum Lagrange cost. This implies that for each existing MB partition (subpartition) within the MB, bit rate and distortion are calculated by actually encoding and decoding the video. Therefore, the encoder can achieve the best R-D performance results, at the expense of extra complexity.

For evaluating the R-D-cost, the standard has to obtain the encoding rate $\boldsymbol{R}$ and the *distortion* $\boldsymbol{D}$ of each MB partition (sub-MB partition). The former is obtained by first computing the difference between the original MB and its predictor. Thereafter, a 4×4 Hadamard transform (HT) has to be applied followed by a quantization process. The distortion, $\boldsymbol{D}$, is obtained by performing an inverse quantization process followed by its inverse HT and then comparing the original MB to the reconstructed one. The H.264 standard then chooses the MB mode with the least cost, $\boldsymbol{J}$. The cost is evaluated using the Lagrange function $J = D + \lambda \times R$, where λ is the Lagrange multiplier, which is given by $\lambda = 0.85 \times 2^{(\mathrm{QP}-12)/3}$, where QP is the MB quantization parameter. Fig. 3 depicts the overall process.

One of the main drawbacks of this method is its excessive computational cost. On the contrary, the encoder can achieve the best R-D performance results. However, for many applications, the use of the Lagrange multiplier may be prohibitive. This is the case when developing a real-time transcoder.

B. The SAE-Cost

In this method, the H.264 JM reference software encoder selects the best MB mode by using the sum of absolute errors (SAE). This implies that for each existing MB partition (subpartition) within the MB, a predictor within the pixel-domain is created from the motion estimation or intra-prediction of the current partition and the SAE cost is evaluated. For each MB and for each color component (Y,Cr,Cb), one prediction mode has to be obtained. The best mode is determined corresponding to the mode exhibiting the minimum SAE cost. One of the main advantages of this method is its low computational cost. On the contrary, the R-D performance results are suboptimal.

C. Fast Motion Estimation (FME) Option

Motion estimation is one of the most important tools in the H.264 encoder for exploiting the high temporal redundancy between successive frames to improve video coding efficiency. Motion estimation is also the most time consuming part in the H.264 encoder (since it is also used for mode decision). Generally motion estimation is conducted in two steps: 1) integer pel motion estimation; and 2) fractional pel motion estimation around the position obtained by the integer pel motion estimation.

Algorithms for fast motion estimation (FME) are currently a hot research topic, especially fast integer pel motion estimation has received much more attention because traditional fractional pel motion estimation only takes a very small proportion of the computation compared with integer pel motion estimation. Fast motion estimation algorithms such as EPZS [15], UMHexagonS [16], and SEA [17] have been proposed to reduce the search space required in motion estimation.

The UMHexagonS algorithm proposed by Tsinghua University has been adopted by the H.264/MPEG-4 Part 10 (AVC) reference software implementation [12]. This algorithm uses the hybrid and hierarchical motion search strategies. It includes four steps making use of different search patterns: 1) predictor selection and prediction mode reordering; 2) unsymmetrical-cross search; 3) uneven multihexagon-grid search; and 4) extended hexagon-based search. When applying the second and third step, the motion estimation accuracy can be nearly as high as the one obtained when conducting a full search. However the computation load and number of operations can be further reduced. Unsymmetrical-cross search uses prediction vector as the search center and extends in the horizontal and vertical directions. Uneven multihexagon-grid search includes two substeps: 1) a full search is carried out around the search center and 2) a 16 multihexagon-grid search strategy is taken. an extended hexagon-based search is used as a center-based search algorithm, including hexagon search and diamond search in a small range. In the H.264 reference software, the FME algorithm (based on the UMHexagonS algorithm) can be employed for the motion estimation in addition to the original full search (FS) algorithm.

D. Observations and Motivation

As mentioned in the previous sections, in the current H.264 JM reference software, in order to encode a given MB in an interframe, the encoder tries all possible prediction modes in the following order; SKIP, Inter16 × 16, Inter16 × 8, Inter8 × 16, Inter8 × 8, (Inter8 × 4, Inter4 × 8, Inter4 × 4), Intra4 × 4, Intra8 × 8 and Intra16 × 16.

It is worth mentioning that each inter-mode decision requires a motion estimation process. This implies that for each MB partition (sub-MB partition) within an MB, motion estimation is done first for all block types and the resulting cost is used for the mode decision. Moreover, although the probability of deciding intra-modes is much less than that of inter-modes in inter-frames, the encoder computes the cost of intra-mode at every MB.

It is obvious that this "try all and select the best" philosophy proves 100% effective in finding the optimal coding mode of MB at the expense of an extremely high computational complexity. The complexity analysis described in [18] shows that examining all possible modes takes the most time out of the total encoding time (the maximum number of computing cost values for MB mode decision is 768).

Based on these observations, we present an innovative approach to jointly optimize the decision mode and motion estimation. In our proposal, the H.264 MB mode computation problem is posed as a data classification problem where the MPEG-2 MB coding mode and residual have to be classified into ***one*** of the several H.264 coding modes. Unlike the reference software where MVs are estimated for all inter-mode block types, in our approach, no motion estimation is required for a particular mode if that mode is not selected by the mode decision algorithm. In the same way, no intra-prediction is required for a particular inter-MB if that mode is not selected by the mode decision algorithm. The proposed approach will perform the motion estimation only for the final MB mode determined by the decision tree.

III. Relevant Proposals to Speed-Up the Inter-Frame Prediction

Due to the fact that the inter-frame prediction in H.264 is the most computationally expensive process, several fast inter-mode decision algorithms have been proposed in the literature. More recently, fast mode decision algorithms with a focus on MPEG-2/H.264 transcoding have being also proposed in the literature. In general, these algorithms achieve significant time saving with negligible loss of coding efficiency. In the following, we introduce some of the most prominent ones.

Lim *et al.* [19] proposed a fast inter-mode selection algorithm using the information supplied by the intra-prediction mode step and an edge map. Consequently, the process of intra-prediction is performed first. However, due to the small probability of intra-modes in inter-frames in real video sequences (maximum 9% and average of 3%, [20]) suggests that the current practice of deciding the best intra-mode first and subsequent decision of an inter-mode may have a certain limit in reducing the computational complexity.

Based on this observation, Jeon *et al.* [21] proposed a new mode decision method based on "selective intra-mode decision" that investigates intra-modes after deciding the best inter-mode. Furthermore, this algorithm investigates various intra-modes only when it is believed to be certainly worthwhile. The proposed algorithm provides considerable reduction in the computational complexity at the expense of a small coding loss.

Kim *et al.* [22] proposed an adaptive mode decision algorithm using the properties of an all-zero coefficients block that is produced by quantization and coefficient thresholding to effectively eliminate unnecessary inter-modes. As a result, the proposed algorithm is two times faster than the H.264 reference software and shows better performance than the other fast algorithms mentioned above [19], [21]. However in this method, there are several threshold values that should be predefined. Furthermore, the transform coefficients must be available to make a fast decision.

Recently, a fast inter-mode decision algorithm for H.264 video coding has been proposed by Wu *et al.* in [23]. This algorithm makes use of the spatial homogeneity of a video object's textures and the temporal stationary characteristics inherent in video sequences. However, the method suffers from a drawback since it requires an edge image for the texture information and a difference image for the temporal stationary characteristics.

More recently, fast mode decision algorithms for H.264 with a focus on MPEG-2/H.264 transcoding have being proposed in the literature. In [24] an efficient block size mode selection algorithm for the variable-size block-matching in the MPEG-2 to H.264 transcoding has been presented by Chen *et al.* Depending on leveraging the available motion information carried by the MPEG-2 bit-streams, the proposed algorithm is used to determine which one of the block size modes should be used for each MB. The simulation results showed that the performance of the proposed algorithm are close to that of a cascaded pixel-domain transcoder when all the block size modes are enabled and the exhaustively full search method is used to determine the best block size modes. However, the whole transcoding time can be only reduced by around 20% on the average while the bit rate is slightly increased.

Lu *et al.* have proposed a fast inter-mode decision for the B- and P-frames and a fast intra-prediction for MPEG-2/H.264 transcoding. In addition, the authors have developed a fast motion estimation algorithm by reusing the motion information from MPEG-2. In their algorithm, the suggested inter-mode prediction for SKIP, 16×16, and DIRECT modes used neighboring MBs of the target MB in the current frame. The proposed MPEG-2/H.264 algorithm provides considerable reduction in computational complexity at the expense of a small coding efficiency loss.

Finally, Kim and Jeon [25] have proposed a fast algorithm for transcoding from MPEG2 to H.264 in the spatial domain. The authors exploit three pieces of information included in an MPEG-2 bit stream: the coded MB type, the coded block pattern and the MV. According to the coded MB type and coded block pattern, the authors adaptively select the MB mode during the H.264 encoding process. Furthermore, the MV is also reused when an inter16×16 mode is selected as a MB mode. Simulation results show that the proposed transcoder dramatically reduces total transcoding time at comparable peak signal-to-noise ratio (PSNR).

IV. Fast MB Mode Decision Using Machine Learning

In this section we present a novel MB mode decision algorithm for inter-frame prediction based on machine learning techniques to be used as part of a very low complexity MPEG-2 to H.264 video transcoder.

Machine learning refers to the study of algorithms and systems that "learn" or acquire knowledge from experiences. Machine learning uses statistics with different kinds of algorithms to solve a problem by studying and analyzing the data. Machine learning has been used in an extensive range of applications including search engines, medical diagnosis, stock market analysis, classifying DNA sequences, speech and handwriting recognition, object recognition in computer vision, game playing and robot motion. The machine learning techniques used in this work are based on supervised learning. Supervised learning uses data samples with known measurements and class membership to create a set of rules to classify data samples with known measurements and assign class membership. Deductive machine learning deduces new rules/knowledge from existing rules and inductive machine learning uses the analysis of data sets for creating a set of rules to take decisions. These rules can be used, in the machine learning, to build a decision tree using a set of experiments or examples, named the training data set. This set of data must have the following properties [26].

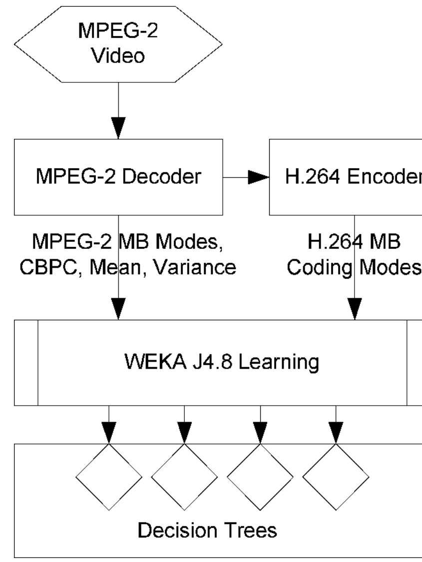


Fig. 4. Process for building decision trees for MPEG-2 to H.264 transcoding.

1) Each attribute or variable can take nominal or numerical values, but the number of attributes cannot vary from one sample to another. That is to say, all the samples in the training data set used for training the model must have the same number of variables.
2) The set of categories that the samples can be assigned must be known *a priori* to enable supervised learning.
3) The set of categories must be finite and must be different from one another.
4) Since inductive learning consists in obtaining generalization from samples, it is assumed that a sufficient large number of examples exist.

In this paper, we apply machine learning principles by building a decision tree enabling the development of a very low complexity transcoding mechanism. The decision tree is used to determine the coding modes of the P-frames MBs of the output H.264 encoded video sequences. The coding modes are determined by using the information gathered during the MPEG-2 decoding stage. Fig. 4 depicts the process of building the decision trees for determining the modes of the MBs during the MPEG-2 to H.264 transcoding process. During the decoding process of the input MPEG-2 video, the MB coding mode, the coded block pattern (CBPC), and the mean and variance of the residual information for this MB (calculated for its 4×4 subblocks – resulting in 16 means and 16 variances for each MB) are saved. Since MPEG-2 uses 16×16 motion compensation (MC) and does not temporally decorrelate an image in full, the MC residual can thus be exploited to understand the temporal correlation of variable block sizes in H.264. The decoded MPEG-2 video is then encoded using a standard H.264 encoder. The coding mode of the corresponding MBs in H.264 is also saved. Based on the MPEG-2 data and the corresponding H.264 coding mode decision for each MB, a machine learning algorithm by means of decision trees is created to classify each MB into one of the several H.264 MB coding modes.

A. Creating the Training Files

A decision tree is constructed by mapping the observations about a data set to a tree made of arcs and nodes. The nodes are the variables and the arcs are the possible values for that variable. A decision tree can have more than one level; in that case, the nodes (leafs of the tree) represent the decisions based on the values of the different variables. These types of trees are used in the machine learning processes for discovering the relationships in a data set. The tree leafs are the classifications and the branches are the features that lead to a specific classification. A decision tree is a classifier based on a set of attributes allowing us to determine the category of an input data sample.

The decision tree for the transcoder developed herein was created using the WEKA data mining tool [26]. The files used by the WEKA data mining program are known as Attribute-Relation File Format (ARFF) files. An ARFF file is written in ASCII text and contain the existing relationship between a set of attributes. An ARFF file has two different sections: 1) the header which contains the name of the relation, the attributes that are used, and their types and 2) the section containing the data.

The training sets were developed by using MPEG-2 sequences encoded at a higher quality than the typical broadcast encoding rates since no B-frames have been used. The H.264 decisions in the training set were obtained from encoding the MPEG-2 decoded sequence with a QP of 25 and the R-D optimization option enabled. After extensive experimentation, we found that sequences containing regions varying from homogenous to high-detail serve as good training sets, such as the widely used *Flower* and *Football* video sequences. The goal has been to develop a single, generalized, decision tree to be used for the MPEG-2 to H.264 transcoding process. We first found that training based on the *Flower* sequence was sufficient to make accurate decisions for a large number of videos. We then tested the performance of the decision tree on nine other sequences.

Fig. 5 shows the decision trees built using the process depicted in Fig. 4. As shown in Fig. 4, the *Decision Tree* for the proposed transcoder is a hierarchical decision tree consisting of three different WEKA trees or classifiers, namely: 1) a classifier for Intra, Skip, Inter 16×16, and Inter 8×8 MBs; 2) a classifier for mapping an inter-16×16 MB into one 16×16, 16×8, or 8×16 MB; and 3) a classifier for mapping an inter-8×8 MB into one of 8×8, 8×4, 4×8, or 4×4 MB . This paper focuses on computing the mode of inter-MBs of the P-frames. The classification and processing of the intra-MBs is out of the scope of this article.

For creating the first WEKA tree (Fig. 5 Node 1), the *first training data set* uses the mean and variance of each one of the sixteen 4×4 residual subblocks, the MB mode in MPEG-2 (Skip, intra, and three nonintra-modes, labeled as 0, 1, 2, 4 and 8 in the code shown below), the coded block pattern (CBPC) used in MPEG-2, and the corresponding H.264 MB coding mode decision for that MB as determined by the standard reference software.

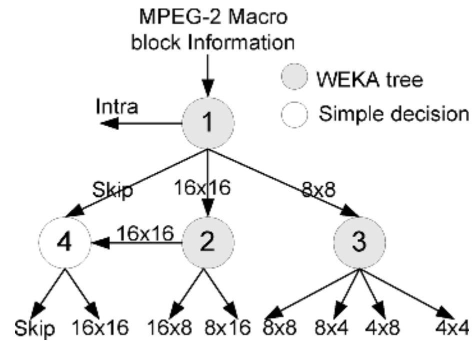


Fig. 5. Decision tree.

The *second training data set*, used for creating the second WEKA tree (Fig. 5 Node 2), was made using the samples (MBs) encoded as 16×16 MBs in the H.264 reference encoder. It uses the mean and variances of each one of the sixteen 4×4 residual subblocks, the MB mode in MPEG-2 (in this case only the three nonintra-modes), the coded block pattern (CBPC) in MPEG-2, and the corresponding H.264 MB coding submode decision in the 16×16 mode, as determined by the standard reference software: 16×16, 16×8 or 8×16. This WEKA tree determines the final coding mode of the MBs classified as inter-16×16 by the first tree.

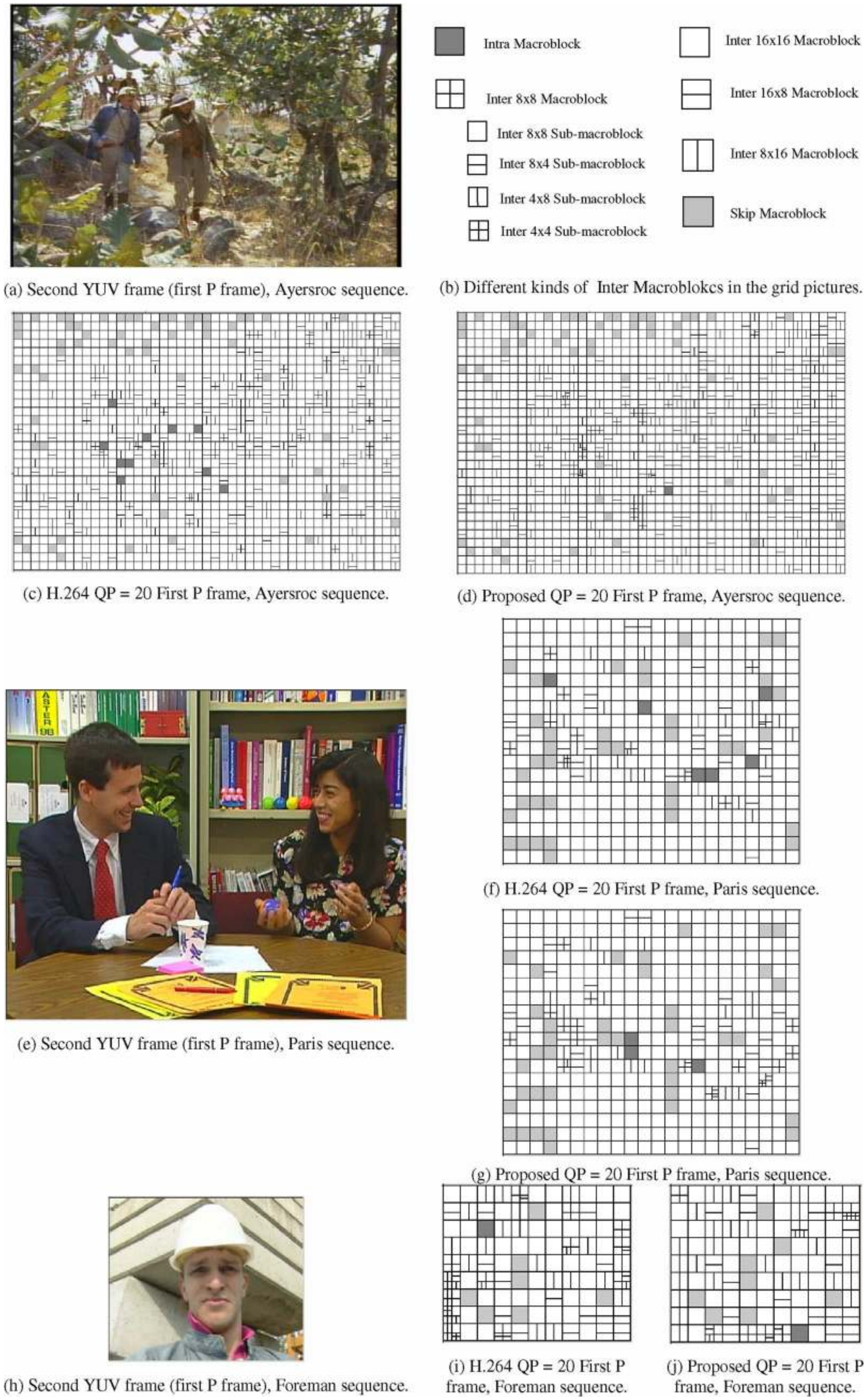
The *third and last training data set*, was used to create the third WEKA tree (Fig. 5 Node 3) and was built using the samples (MBs) encoded as inter-8×8 MBs in the H.264 reference encoder. It uses four means and four variances of 4×4 residual subblocks, the MB mode in MPEG-2 (the three nonintra-modes), the coded block pattern (CBPC) in MPEG-2, and the corresponding H.264 MB subpartition decision in the 8×8 mode, as determined by the standard reference software: 8×8, 8×4, 4×8 or 4×4. Since this decision is made separately for each 8×8 subblock, only the four means and four variances of the 4×4 residual subblocks are used in each sample for training the model.

Based on these training files, the J48 algorithm implemented in the WEKA data mining tool was used to create the WEKA decision trees. The J48 algorithm is an implementation of the C4.5 algorithm proposed by Ross Quinlan [27]: the algorithm widely used as a reference for building decision trees.

The decision tree, that is proposed to solve the inter-prediction problem, is a model of the data that encodes the distribution of the class label in terms of the attributes. The final goal of this decision tree is to help find a simple structure to show the possible dependences between the decoded MPEG-2 data and the H.264 MB coding modes.

B. Decision Tree

Fig. 5 shows the decision tree used in the proposed transcoder. The decision tree consists of three WEKA decision trees, shown in Fig. 5 with grey balls. The first WEKA tree is used to check for the skip, Intra, 8×8 and 16×16 MBs modes. If an MB is 8×8 or 16×16, a second and a third decision tree is used for selecting the final coding mode of the MB. The WEKA tool determined the mean and variance thresholds for each of the three



(a) Second YUV frame (first P frame), Ayersroc sequence.
(b) Different kinds of Inter Macroblokcs in the grid pictures.
(c) H.264 QP = 20 First P frame, Ayersroc sequence.
(d) Proposed QP = 20 First P frame, Ayersroc sequence.
(e) Second YUV frame (first P frame), Paris sequence.
(f) H.264 QP = 20 First P frame, Paris sequence.
(g) Proposed QP = 20 First P frame, Paris sequence.
(h) Second YUV frame (first P frame), Foreman sequence.
(i) H.264 QP = 20 First P frame, Foreman sequence.
(j) Proposed QP = 20 First P frame, Foreman sequence.

Fig. 6. MB mode decisions generated by the proposed algorithm for the first P-frame in the Ayersroc, Paris, and Foreman sequence.

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 herein should prove the guidelines to develop the decision trees and replicate the experiments.

The decision tree works as follows:

1) Node 1: The inputs for this node are the MPEG-2 coded MBs. In this node, the first WEKA decision tree is used to decide whether the MB should be encoded into the H.264 format. This tree examines the residual level of the MB. The output of this node is a first level decision that classifies the MB as Skip, Intra, Inter-8×8 or Inter-16×16. The intra-decision process is not discussed in this paper. In all the other cases, the algorithm has to make a second level decision taking into account the outcome of the first decision. For example, the following rules were given by WEKA.

- If the MPEG-2 MB has been encoded as "MC not coded," (nonzero MV present, none of the 8×8 block has coded coefficients), then the MB will be coded as 16×16 in H.264. A second decision level will be made to select the best choice in this case (see Node 2).
- If the MPEG-2 MB has been encoded as intra-mode, the MB will be encoded as intra-or inter-8×8 mode in H.264. In some cases the algorithm may suggest Intra, and no further processing will take place for that MB. If the algorithm suggests the 8×8 mode, a second level decision will be done (see Node 3).
- If the MPEG-2 MB was encoded in skip mode, then the H.264 decision mode should be skip. The decision takes place in Node 4.

2) Node 2: The inputs for this node are the 16×16 MBs classified by the Node 1. In this node we use again a decision tree created with WEKA to decide whether the MB should be encoded in H.264 as 16×16, 16×8 or 8×16. This tree examines if there are continuous 16×8 or 8×16 subblocks that might result in a better prediction. The output of this node is the 16×16 submode decision used for coding the MB: 16×16, 16×8 or 8×16. When the node decision is 16×8 or 8×16, the coding mode is decided for the MB. In other case, the evaluation continues in Node 4, where the final decision will be made.

3) Node 3: The inputs for this node are the MBs classified by Node 1 as 8×8. This node evaluates only the H.264 8×8 modes using the third WEKA tree and selects the best option: 8×8, 8×4, 4×8 or 4×4. As explained in the previous section, this tree is run four times, once for each of the four sub-MBs in the MB. This node is different from the others since this one only uses four means and four variances to make the decision.

4) Node 4: The inputs for this node are skip-mode MBs in the MPEG-2 bit stream classified by Node 1, or the 16×16 MBs classified by Node 2. This node evaluates only the H.264 16×16 mode (without the submodes 16×8 or 8×16). Then, the node selects the best option, skip or inter-16×16.

Since the MB mode decision, and hence the thresholds used in the decision trees, depend on the QP used in the H.264 encoding stage, the mean and variance threshold will have to be different at each QP. The two possible solutions here are: 1) to develop the decision trees for each QP and use the appropriate decision tree depending on the QP selected and 2) to develop a single decision tree and adjust the mean and variance threshold used by the trees based on the QP. The first option is complex since implies to switch between 52 different decision trees resulting in 156 WEKA trees for a transcoder. Since the QP used by H.264 is designed to change the quantization step size and the relationship between the QPs is well defined, this relationship can be used to adjust the mean and variance thresholds. The proposed transcoder uses a single decision tree developed for a midQP of 25 and then adjusted for other QPs. Since the quantization step size in H.264 doubles when QP increases by 6, the 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 QP values lower than 25 thresholds are proportionally increased.

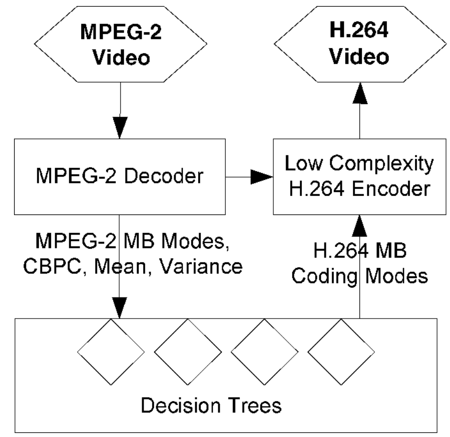


Fig. 7. Proposed transcoder.

Fig. 6 shows examples of the results obtained by applying our proposed algorithm. Fig. 6 shows the difference between the MB mode decisions made by the H.264 standard (with the R-D-optimized option enabled), and the proposed algorithm, with a value of 20 for QP. From these figures, it is clear that the proposed algorithm obtains very similar results to those obtained using the full estimation of the H.264 standard.

V. Performance Evaluation

The proposed low complexity MB coding mode decision algorithm has been implemented in the H.264/AVC reference software, version JM 10.2 [12]. Fig. 7 shows the overall operation of the proposed transcoder. The MPEG-2 video is decoded and the information required by the decision trees is gathered in this stage. The additional processing implies obtaining the mean and variance of the 4×4 subblocks of the residual MBs. The MB coding mode decision determined by the decision trees is used in the low complexity H.264 encoding stage. This is an H.264 reference encoder with the MB mode decision replaced by a simple mode assignment process built around a decision tree. The H.264 video encoder takes as input the decoder MPEG-2 video (pixel data) and the MB mode decision from the decision tree and encodes the H.264 video. The MPEG-2 MVs are not used and the encoder performs the motion estimation just for the final MB mode determined by the decision tree.

We have conducted an extensive set of experiments with videos representing a 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 transcoder is a high quality MPEG-2 video. Since the proposed transcoder addresses transcoding P-frames in MPEG-2 to H.264 P-frames, MPEG-2 bit streams were created without B frames.

TABLE I
BIT RATES FOR THE INPUT SEQUENCES

Format	Bitrate
CCIR-601 (720x480)	5 Mbps
CIF (352x288)	1.15 Mbps
QCIF (176x144)	0.768 Mbps

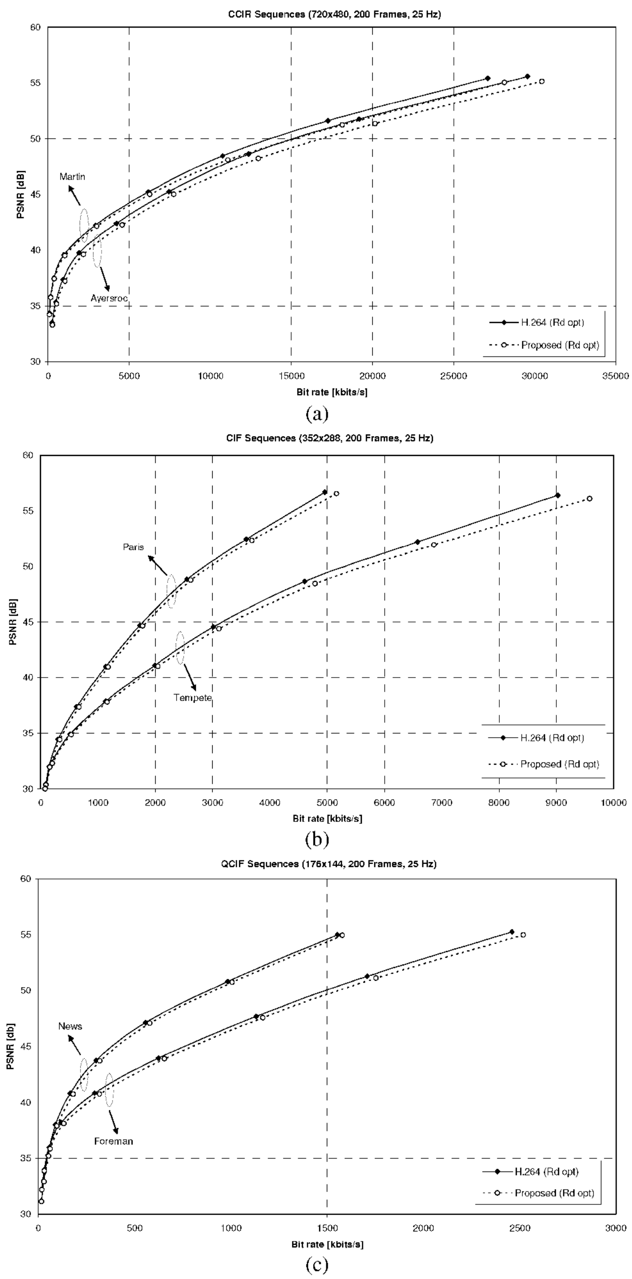


Fig. 8. R-D results for R-D-cost without FME option.

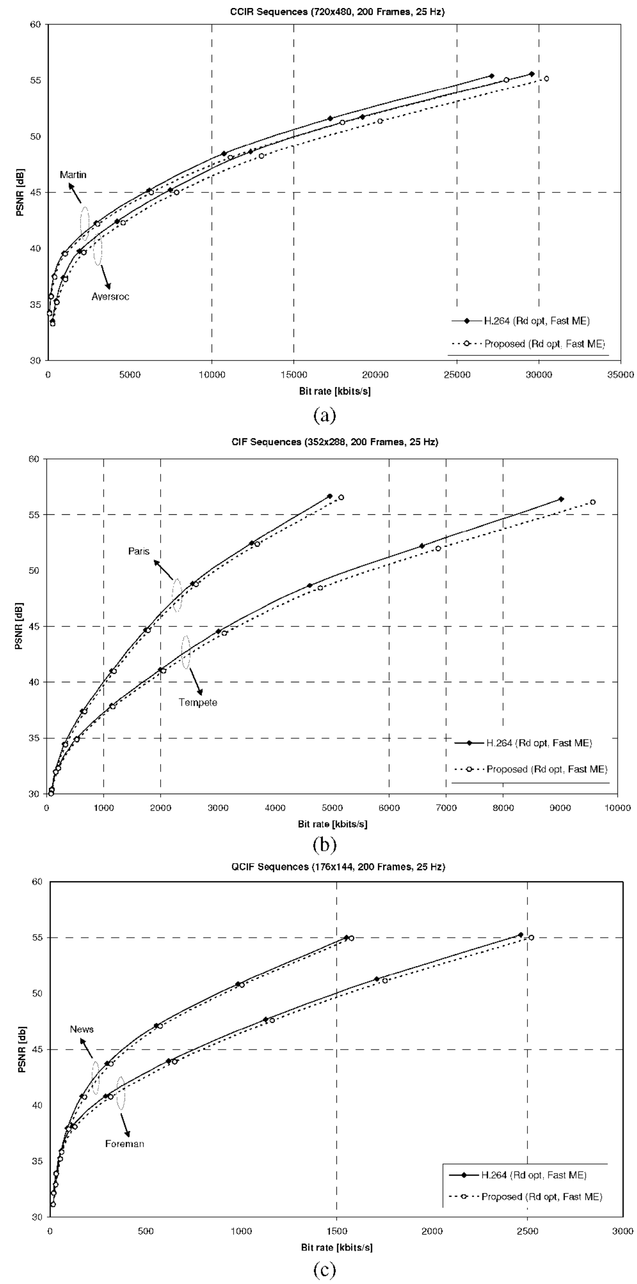


Fig. 9. R-D results for R-D-cost with FME option.

Since the B frames, which are much smaller than P-frames, are not used in the input video, the video has to be encoded at higher than the typical encoding rates for equivalent broadcast quality. Table I shows the bit rates used for the input MPEG-2 video.

The sequences have been encoded with H.264 using the QP factors ranging from 5 up to 45 in steps of 5. This corresponds to the H.264 QP range used in most practical applications. The size of the GOP is 12 frames; where the first frame of every GOP was encoded as I-frame, and the rest of the frames of the GOP were encoded as P-frames. The rate control and CABAC algorithms were disabled for all the simulations. The number of reference in P-frames was set to 1 and the motion search range was set to ±16 pels with a MV resolution of $1/4$ pel. The ProfileIDC was set to high for all the simulations, with the FRExt options enabled. The simulations were run on a P4 HT at 3.0-GHz Intel machine with 512 MB RAM. Since the training has been based on R-D optimized MB mode decisions, the proposed algorithms always disable the R-D optimization in the H.264 encoding stage and still perform substantially better than SAE based reference transcoder. The experiments have shown that the proposed approach performs extremely well across all bit rates and resolutions. The results are reported for six different sequences: two for each of the three resolutions shown in Table I.

TABLE II
COMPARISON OF MB MODE DECISION ALGORITHMS (H.264[R-D-COST] VERSUS PROPOSED)

	RD-cost without FME option			RD-cost with FME option		
Sequence	ΔTime (%)	ΔPSNR (dB)	ΔBitrate (%)	ΔTime (%)	ΔPSNR (dB)	ΔBitrate (%)
Martin	-80,19	-0,338	9,49	-92,68	-0,389	11,31
Ayersroc	-78,82	-0,506	13,99	-90,18	-0,531	14,62
Paris	-82,00	-0,277	5,64	-95,30	-0,274	5,56
Tempete	-81,14	-0,292	5,75	-93,41	-0,304	6,00
Foreman	-81,95	-0,326	8,67	-95,15	-0,309	8,15
News	-81,27	-0,302	7,33	-95,53	-0,287	6,98
Average	**-80,90**	**-0,340**	**8,47**	**-93,70**	**-0,349**	**8,77**

The performance of the proposed very low complexity transcoder is compared with a reference transcoder comprised of a full MPEG-2 decoder followed by a full H.264 encoder. We compare the performance of our proposal to the full H.264 encoder when the R-D-cost (with and without FME option enabled) and the SAE-cost (with and without FME option enabled) are used. The metrics used to evaluate the comparative performance are: the R-D function, the difference of coding time ($\Delta\mathrm{Time}$), the PSNR difference ($\Delta\mathrm{PSNR}$) and the bit rate difference ($\Delta\mathrm{Bitrate}$). We show the results in the R-D function for a complete range in terms of output bit rates, in order to demonstrate that the proposed approach performs extremely well across all bit rates of H.264. The averaged PSNR values of luma (Y) and chroma (U,V) is used in the R-D function graphs. The computation of the averaged PSNR is based on the following equation:

$$\overline{\mathrm{PSNR}} = \frac{4 \times \mathrm{PSNR}_Y + \mathrm{PSNR}_U + \mathrm{PSNR}_V}{6}.$$

In order to evaluate the timesaving of the fast MB mode decision algorithm, the following calculation is defined to find the time differences. Let T_{JM} denote the coding time used by the H.264/AVC reference software encoder (version JM 10.2) and T_{FI} be the time taken by the fast MB mode decision algorithm proposed, and $\Delta\mathrm{Time}$ is defined as:

$$\Delta\mathrm{Time}(\%) = \frac{T_{\mathrm{FI}} - T_{JM}}{T_{JM}} \times 100.$$

The PSNR and bit rate differences are calculated according to the numerical averages between the R-D-curves derived from JM 10.2 encoder and the fast MB mode decision methods, respectively. The detail procedures in calculating these differences can be found in the JVT document authored by Bjontegaard [28], used as reference by the JVT Test Model Ad Hoc Group [29]. Note that the PSNR and bit rate differences should be regarded as equivalent, i.e., a larger (smaller) PSNR difference implies a larger (smaller) bit rate difference.

Fig. 8 shows the R-D results for the reference and proposed transcoder with the R-D optimization option being enabled and the FME option disabled. Fig. 9 shows the R-D results for the reference and proposed transcoder with the R-D optimization enabled and FME option enabled. As seen from the figures, the PSNR-bit rate obtained with the proposed transcoder deviates slightly from the results obtained when applying the considerable more complex reference transcoder.

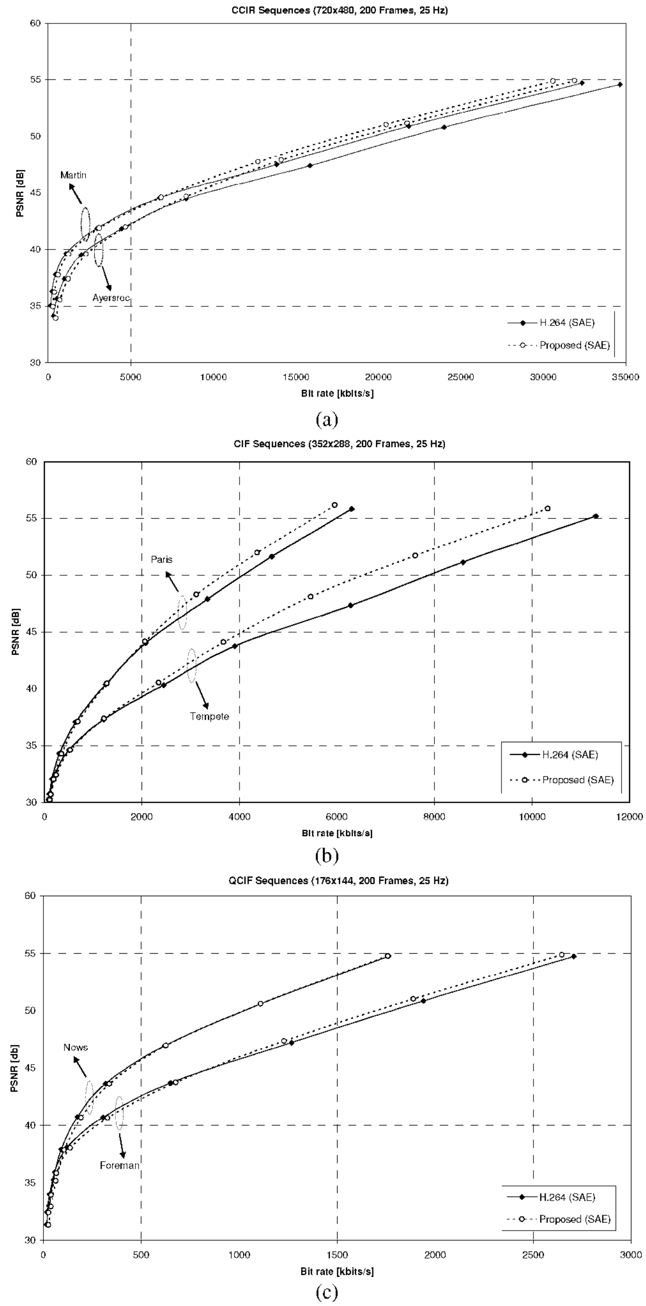
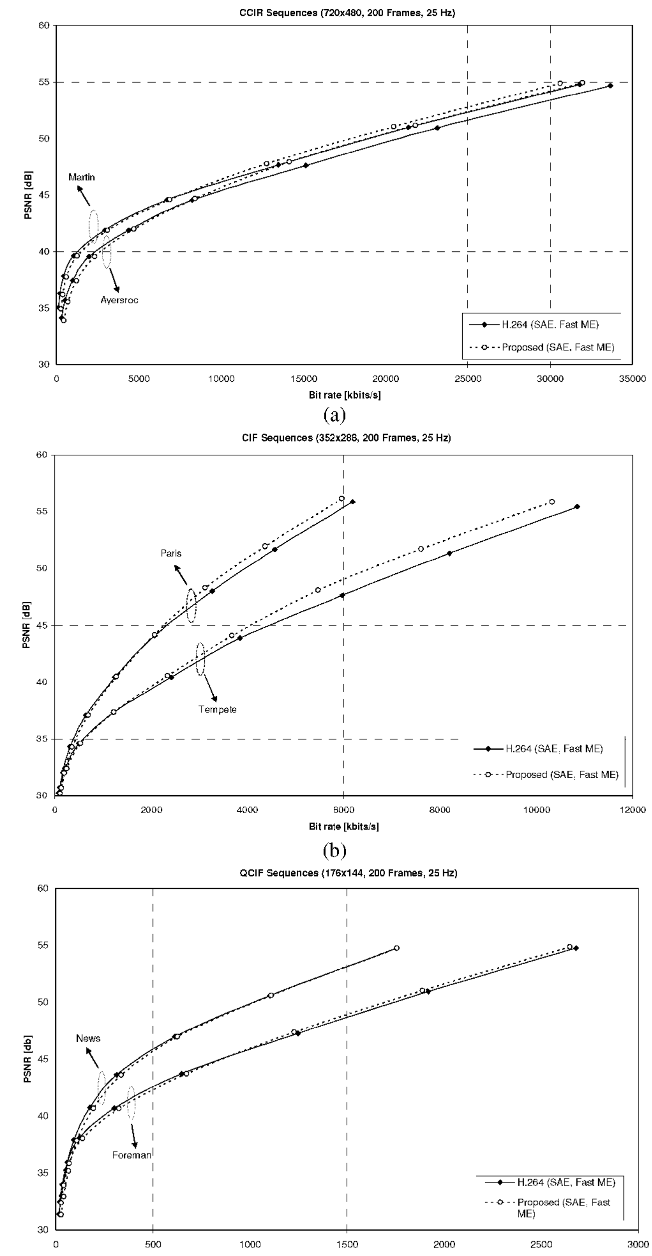


Fig. 10. R-D results for SAE-cost without FME option.

Table II shows the results in terms of $\Delta\mathrm{Time}$, $\Delta\mathrm{PSNR}$, and $\Delta\mathrm{Bitrate}$. Compared with the reference transcoder, the proposed transcoder has a PSNR drop of at most 0.3 dB for a given bit rate and a bit rate increase of at most 8% for a given PSNR. This negligible drop in R-D performance is more than offset by the reduction in computational complexity. As shown in Table II, the proposed transcoder reduces the inter-frame prediction time by over 80% with R-D optimization, and more than 93% with FME enabled.

Fig. 10 shows the R-D results for the reference and proposed transcoder with SAE-Cost (R-D optimization disabled) and FME disabled. Fig. 11 shows the R-D results for the reference and proposed transcoder with SAE-Cost (R-D optimization disabled) and FME enabled. As seen from the figures, in some cases the proposed transcoder shows better results than the ones obtained by the reference transcoder.



(c)

Fig. 11. R-D results for SAE-cost with FME option.

Table III compares our proposal to the SAE-cost approach of the reference software in terms of the time, PSNR and bit rate differences. The computational cost is reduced by over 40% with respect to the one required by the SAE-cost approach with the FME disabled and by over 81% in the case when the FME option is enabled. The proposed transcoder exhibits a PSNR drop of less than 0.02 and 0.12 dB with respect to the ones obtained by the SAE-cost approach without and with the FME option enabled, respectively. The average bit rate difference increase obtained by our proposal is less than 4% and 6% with respect to the bit rate obtained by the SAE-cost approach without and with the FME option enabled, respectively. It is clear that there is a tradeoff between the computational cost of our algorithm and the resulting video quality. It is also clear that the quality reduction is very small and negligible for most practical video applications.

TABLE III
COMPARISON OF MB MODE DECISION ALGORITHMS (H.264[SAE-COST] VERSUS PROPOSED)

	SAE-cost without FME option			SAE-cost with FME option		
Sequence	**ΔTime (%)**	**ΔPSNR (dB)**	**ΔBitrate (%)**	**ΔTime (%)**	**ΔPSNR (dB)**	**ΔBitrate (%)**
Martin	-43,60	-0,165	9,42	-81,91	-0,309	13,39
Ayersroc	-44,63	0,099	2,35	-81,55	-0,081	6,20
Paris	-38,98	0,051	0,80	-80,85	-0,031	2,20
Tempete	-40,50	0,507	-7,32	-81,72	0,259	-4,00
Foreman	-40,00	-0,174	6,63	-82,35	-0,208	7,00
News	-38,67	-0,408	10,75	-80,00	-0,334	11,02
Average	**-41,06**	**-0.015**	**3.77**	**-81.39**	**-0.117**	**5,96**

TABLE IV
MEAN ENCODING TIME (MILLISECONDS) PER FRAME WITH THE REFERENCE TRANSCODER

Sequence	RD Opt	RD Opt + FME	SAE	SAE + FME
Martin	7370	6420	2110	940
Ayersroc	7650	6820	2095	1030
Paris	2305	2020	590	235
Tempete	2360	2050	605	290
Foreman	565	495	155	68
News	550	470	150	55

TABLE V
MEAN ENCODING TIME (MILLISECONDS) PER FRAME WITH THE PROPOSED TRANSCODER

Sequence	RD Opt	RD Opt + FME	SAE	SAE + FME
Martin	1460	470	1190	170
Ayersroc	1620	670	1160	190
Paris	415	95	360	45
Tempete	445	135	360	53
Foreman	102	24	93	12
News	103	21	92	11

Tables IV and V show the mean encoding time (milliseconds) per frame for the reference transcoder and the proposed transcoder, respectively. Based on the results shown in Tables IV and V, the proposed transcoder with the SAE and FME options enabled, labeled SAE+FME in the tables, outperforms all the other configurations.

The proposed transcoder with the R-D optimization and FME options enabled is still faster than the fastest case of the reference transcoder (SAE + FME). It is clear that the use of the FME option reduces substantially the complexity of the transcoder. When using the R-D optimization procedure with the proposed transcoder doubles the complexity compared to the SAE+FME case. The decision to enable or disable the R-D optimization can be based on the operating bit rates and sensitivity to the PSNR drop as dictated by the video application requirements. At higher bit rates, the R-DOPT + FME option results on a 0.6 dB gain with respect to the PSNR obtained when using the SAE + FME option, i.e., a gain of 0.6 dB can be obtained by doubling the complexity. However, at lower bit rates, the PSNR gain reduces to about 0.3 dB.

Finally, Table VI shows the quality variation versus time reduction of the proposed transcoder with respect to the reference

TABLE VI
QUALITY VARIATION VERSUS TIME REDUCTION (FOR TRANSCODING RATE)

Sequence	MPEG-2 Bit Rate (Mbps)	Quality Variation from Reference Transcoder (dB)				Time Reduction from Reference Transcoder (%)			
		RD OPT	RD FME	SAE	SAE FME	RD OPT	RD FME	SAE	SAE FME
Ayersroc	5.0	- 0.3	- 0.3	0.0	- 0.1	80.0	90.5	43.3	82.3
Martin	5.0	- 0.2	- 0.2	- 0.1	- 0.1	80.5	92.8	42.1	82.0
Tempete	1.15	- 0.2	- 0.2	0.0	0.0	80.0	93.8	41.1	82.5
Paris	1.15	- 0.3	- 0.3	0.0	- 0.1	81.6	95.6	38.5	80.7
Foreman	0.768	- 0.3	- 0.3	0.0	0.0	83.5	95.5	37.4	82.6
News	0.768	- 0.2	- 0.2	0.0	0.0	84.1	96.0	35.1	81.1

transcoder for the same input bit rates shown in Table I. The results in Table VI show a reduction of up to 96% in the computational complexity when using our proposed scheme. Furthermore, using the proposed transcoder reduces the PSNR up to 0.3 dB when the R-D optimization option is enabled and by up to 0.1 dB when using the SAE-cost based transcoder. Our results show that our proposed scheme is able to maintain a good picture quality while considerably reducing the number of operations to be performed.

A. Comparison of Different Fast MB Mode Decision Algorithms

As mentioned in Section III, recently various fast MB mode decision algorithms for inter-frame prediction for H.264/AVC video have been proposed in the literature. In this section, we undertake a comparative analysis of our proposal to some of the most relevant algorithms [19]–[25]. In these experimenst, although the test conditions are not exactly the same to the ones reported in the literature, an objective comparison is still possible since all the algorithms follow Bjontegaard and Sullivan's common test rule [28], [30]. The comparison metrics were produced and tabulated based on the difference on the coding time ($\Delta\mathrm{Time}$), the PSNR difference ($\Delta\mathrm{PSNR}$) and the bit rate difference ($\Delta\mathrm{Bitrate}$). The common encoding parameters are as follows: the size of the GOP has been fixed to12 frames; where the first frame of every GOP has been encoded as I-frame, and the remaining frames of the GOP were encoded as P-frames. The R-D optimization option was enabled and the FME option was set to a resolution of ¼ pel. The video sequences used were Mobile (CIF), Paris (CIF), Foreman (QCIF), and News (QCIF). It should be noted that we have selected the same sequences used in [19]–[25]. The results are tabulated in Table VII.

As shown in Table VII, the performance of our fast MB mode decision algorithm for the inter-frame prediction in terms of time saving, which is a critical issue in video transcoding applications, achieves the best results, with a negligible loss on the video quality (< 0.3 dB), and with a slight increment in bit rate with respect to the other methods. This is due to the fact that the proposed approach reduces the H.264 MB mode computation process into a decision tree lookup with very low complexity and the transcoder performs the fast motion estimation just for the final MB mode determined by the decision tree. Furthermore the proposed transcoder can be implemented easily since it only requires the mean and variance of the MPEG-2 residual and a set of rules to compare the mean and variance against a threshold.

TABLE VII
COMPARISON OF DIFFERENT FAST MB MODE DECISION ALGORITHMS

Sequence	Method	ΔTime (%)	ΔPSNR (dB)	ΔBitrate (%)
Mobile	**Our Proposal**	**- 95,03**	**-0,16**	**3,49**
	Lim´s algorithm	- 10,96	-0,01	0,01
	Jeon´s algorithm	- 22,83	-0,02	0,07
	Kim´s algorithm	- 46,93	-0,01	0,10
	Wu´s algorithm	- 9,97	-0,01	0,13
	Lu´s algorithm	-87,60	-0,18	3,86
	Kim&Jeong´s algorithm	-55,28	-0.01	-----
Foreman	**Our Proposal**	**-95,15**	**-0,30**	**8,15**
	Lim´s algorithm	-27,03	-0,06	0,93
	Jeon´s algorithm	-20,24	-0,07	1,16
	Kim´s algorithm	-50,45	-0,02	0,38
	Wu´s algorithm	-25,18	-0,06	1,28
	Chen´s algorithm	-22,00	-0,13	3,00
	Lu´s algorithm	-89,90	-0,26	6,42
	Kim&Jeong´s algorithm	-52,89	-0,04	-----
Paris	**Our Proposal**	**-95,30**	**-0,27**	**5,56**
	Lim´s algorithm	-34,37	-0,04	0,73
	Jeon´s algorithm	-34,69	-0,05	0,92
	Kim´s algorithm	-62,05	-0,02	0,36
	Wu´s algorithm	-31,90	-0,04	0,87
News	**Our Proposal**	**-95,53**	**-0,28**	**6,98**
	Lim´s algorithm	-46,88	-0,05	0,59
	Jeon´s algorithm	-36,11	0,00	0,01
	Kim´s algorithm	-63,70	-0,01	0,20
	Wu´s algorithm	-42,62	-0,07	1,18
	Chen´s algorithm	-23.90	-0,15	2,40

VI. CONCLUSION

In this paper, we have proposed a novel MB mode decision algorithm for inter-frame prediction to be used as part of a high-efficient MPEG-2 to H.264 transcoder. The proposed algorithm uses machine learning techniques to exploit the correlation in the MPEG-2 MC residual and the H.264 coding modes. The WEKA tool was used to develop decision trees for H.264 coding mode decision. The proposed algorithm has very low complexity as it only requires the mean and variance of the MPEG-2 residual and a set of rules to compare the mean and variance against a threshold. The proposed transcoder uses a single decision tree with adaptive thresholds based on the QP selected in the H.264 encoding stage. The proposed transcoder was evaluated using MPEG-2 videos at CCIR, CIF, and QCIF resolutions. Our results show that the proposed algorithm is able to maintain a good picture quality while considerably reducing the computational complexity by as much as 95%. The reduction in computational cost has negligible impact on the quality of the transcoded video. The results show that the proposed transcoder maintains its performance across all resolutions and bit rates. The proposed approach to transcoding is novel and can be applied to develop other transcoders as well. Finally, we have carried out a comparative study with some of the most prominent fast inter-prediction methods for H.264 presented in the literature. The results have shown that the proposed approach achieves the best results for video transcoding applications.

Our future plans will focus on further reducing the complexity of the proposed transcoder by reusing the MPEG-2 MVs followed by a MV refinement. By reusing the MV, we believe, real-time transcoding of CIF resolution video at 30 FPS is within reach.

References

[1] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264, 2003.

[2] *Main Results of the AVC Complexity Analysis.* Implementation Studies Group, MPEG Doc. N4964, ISO/IEC JTC11/SC29/WG11, 2002.

[3] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Trans. Multimedia*, vol. 2, pp. 101–110, Jun. 2000.

[4] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Process. Mag.*, vol. 20, no. 2, pp. 18–29, Mar. 2003.

[5] H. Kalva, A. Vetro, and H. Sun, "Performance optimization of the MPEG-2 to MPEG-4 video transcoder," in *Proc. SPIE Conf, Microtechnol, New Millennium, VLSI Circuits Syst.*, May 2003, pp. 341–350.

[6] S. Dogan, A. H. Sadka, and A. M. Kondoz, "Efficient MPEG-4/H.263 video transcoder for interoperability of heterogeneous multimedia networks," *Electron. Lett.*, vol. 35, no. 11, pp. 863–864, May 1999.

[7] H. Kalva, "Issues in H.264/MPEG-2 video transcoding," in *Proc. Consumer Commun. Netw. Conf.*, Jan. 2004, pp. 657–659.

[8] Y. Su, J. Xin, A. Vetro, and H. Sun, "Efficient MPEG-2 to H.264/AVC intra-transcoding in transform-domain," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 1234–1237.

[9] B. Petljanski and H. Kalva, "DCT domain intra-MB mode decision for MPEG-2 to H.264 transcoding," in *Proc. ICCE 2006*, Jan. 2006, pp. 419–420.

[10] Y.-K. Lee, S.-S. Lee, and Y.-L. Lee, "MPEG-4 to H.264 transcoding using macroblock statistics," in *Proc. ICME 2006*, Jul. 2006, pp. 57–60.

[11] X. Lu, A. M. Tourapis, P. Yin, and J. Boyce, "Fast mode decision and motion estimation for H.264 with a focus on MPEG-2/H.264 transcoding," in *Pro. 2005 IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 1246–1249.

[12] *Reference Software to Committee Draft*. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-F100 JM10.2, 2006.

[13] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[14] T. Wiegand *et al.*, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[15] A. M. Tourapis, O. C. Au, and M. L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 934–947, Oct. 2002.

[16] Z. Chen, P. Zhou, and Y. He, "Fast integer pel and fractional pel motion estimation for JVT," presented at the JVT of ISO/IEC MPEG& ITU–T VCEG, 6th Meeting, Dec. 2002.

[17] M. Yang, H. Cui, and K. Tang, "Efficient tree structured motion estimation using successive elimination," *Proc. IEE, Vision, Image Signal Process.*, vol. 151, no. 5, pp. 369–377, Oct. 2004.

[18] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, 2004.

[19] K. P. Lim *et al.*, *Fast Inter Mode Selection.* JVT of MPEG and VCEG, Doc. JVT-I020, 2003.

[20] J. Lee and B. Jeon, "Pruned mode decision based on variable block sizes motion compensation for H.264," in *Lecture Notes Comp. Sci.*, Nov. 2003, vol. 2899, pp. 410–418.

[21] B. Jeon and J. Lee, Fast Mode Decision for H.264 JVT of MPEG and VCEG, Doc. JVT-J033, 2003.

[22] Y.-H. Kim, J.-W. Yoo, S.-W. Lee, J. Shin, J. Paik, and H.-K. Jung, "Adaptive mode decision for H.264 encoder," *Electron. Lett.*, vol. 40, no. 19, pp. 1172–1173, Sep. 2004.

[23] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko, "Fast intermode decision in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 953–958, Jul. 2005.

[24] G. Chen, Y.-D. Dang, S.-X. Lin, and F. Dai, "Efficient block size selection for MPEG-2 to H.264 transcoding," in *Proc. 2004 ACM Multimedia*, Oct. 2004, pp. 300–303.

[25] D. Kim and J. Jeong, "Fast transcoding algorithm from MPEG-2 to H.264," in *Lecture Notes Comp. Sci.*, 2006, vol. 4319, pp. 1067–1074.

[26] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005.

[27] J. R. Quinlan, *C4.5: Programs for Machine Learning*. : Morgan Kaufmann, 1993.

[28] G. Bjontegaard, "Calculation of average PSNR differences between R-D-curves," presented at the 13th VCEG-M33 Meeting, Austin, TX, Apr. 2001.

[29] *Evaluation Sheet for Motion Estimation.*4 ed. JVT Test Model Ad Hoc Group, 2003.

[30] G. Sullivan and G. Bjontegaard, *Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material.* ITU-T VCEG, Doc. VCEG-N81, 2001.

Gerardo Fernández-Escribano received the M.Sc. degree in computer engineering and the Ph.D. degree from the University of Castilla-La Mancha, Albacete, Spain, in 2003 and 2007, respectively.

In 2004, he joined the Department of Computer Engineering at the University of Castilla-La Mancha, where he is currently a researcher of Computer Architecture and Technology. His research interests include multimedia standards, video transcoding, video compression, video transmission, and machine learning mechanism. He has also been a visiting researcher at the Florida Atlantic University, Boca Raton, and at the Friedrich Alexander Universität, Erlangen-Nuremberg, Germany.

Hari Kalva received the B.Tech. degree in electronics and communications engineering from N.B.K.R. Institute of Science and Technology, S.V. University, Tirupati, India, in 1991., the M.S. degree in computer engineering from Florida Atlantic University, Boca Raton, in 1994, and the Ph.D. and M.Phil. degrees in electrical engineering from Columbia University, New York, in 2000 and 1999, respectively.

He joined the Department of Computer Science and Engineering, Florida Atlantic University ,as an Assistant Professor in August 2003. Prior to that he was a consultant with Mitsubishi Electric Research Laboratories, Cambridge, MA. He was a co-founder and the Vice President of Engineering of Flavor Software, a New York company founded in 1999, that developed MPEG-4 based solutions for the media and entertainment industry. He is an expert on digital audio-visual communications systems with over 12 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. He has over 50 published papers and five patents (11 pending) to his credit. He is the author of one book and coauthor of five book-chapters.

Pedro Cuenca (M'99) received the M.Sc. degree in physics (electronics and computer science, award extraordinary) from the University of Valencia, Valencia, Spain, in 1994, and the Ph.D. degree in computer engineering from the Polytechnic University of Valencia, Valencia, Spain, in 1999.

In 1995, he joined the Department de Computer Engineering at the University of Castilla-La Mancha, Albacete, Spain. He is currently a Full Professor of Computer Architecture and Technology and Vice-Dean of the Escuela Politécnica Superior de Albacete (School of Computer Engineering). He has also been a visiting researcher at The Nottingham Trent University, Nottingham, U.K., and at the Multimedia Communications Research Laboratory, University of Ottawa, Ottawam ON, Canada. His research topics are centered in the area of high-performance networks, wireless LAN, video compression, QoS video transmission and error-resilient protocol architectures. He has published over 80 papers in international Journals and Conferences. He has served in the organization of International Conferences as Chair, Program Co-Chair and Technical Program Comitte member. He is the Chair of the IFIP 6.8 Working Group and a member of the IEEE.

Luis Orozco-Barbosa (M'99) received the B.Sc. degree in electrical and computer engineering from Universidad Autonoma Metropolitana, Mexico, in 1979, the Diplome d'études Approfondies from École Nationale Supérieure d'Informatique et de Mathématiques Appliquées (ENSIMAG), France, in 1984 and the Doctorat de ĺUniversité from Université Pierre et Marie Curie, France, in 1987, both in computer science.

From 1991 to 2002, he was a Faculty member at the School of Information Technology and Engineering (SITE), University of Ottawa, ON, Canada. In 2002, he joined the Department of Computer Engineering at Universidad de Castilla La Mancha, Albacete, Spain. He has also been appointed Director of the Albacete Research Institute of Informatics, a National Centre of Excellence. He has conducted numerous research projects with the private sector and served as Technical Advisor for the Canadian International Development Agency (CIDA). He has published over 180 papers in international Journals and Conferences on computer networks and performance evaluation. His current research interests include Internet protocols, network planning, wireless communications, traffic modeling and performance evaluation.

Antonio Garrido received the degree in physics (electronics and computer science) from the University of Granada. Granada, Spain, in 1986 and the Ph.D. degree from the University of Valencia, Valencia, Spain, in 1991.

In 1986, he joined the Department of Computer Engineering at the University of Castilla-La Mancha, Albacete, Spain, where he is currently a Full Professor of Computer Architecture and Technology and Dean of the Escuela Politecnica Superior de Albacete (School of Computer Engineering). His research interests include high-performance networks, telemedicine, video compression, and video transmission. He has published over 60 papers in international journals conferences on performance evaluation of parallel computer and communications systems and compression and transmission in high-speed networks. He has led several research projects in telemedicine, computer networks and advanced computer system architectures.