

An MPEG-2 to H.264 Video Transcoder in the Baseline Profile

G. Fernández-Escribano, *Member, IEEE*, H. Kalva, *Senior Member, IEEE*, J. L. Martínez, *Student Member, IEEE*,
P. Cuenca, *Member, IEEE*, L. Orozco-Barbosa, *Member, IEEE*, and A. Garrido, *Member, IEEE*

Abstract—Based on our previous efforts, we introduce in this letter a high-efficient MPEG-2 to H.264 transcoder for the baseline profile in the spatial domain. Machine learning tools are used to exploit the correlation between the macroblock (MB) decision of the H.264 video format and the distribution of the motion compensated residual in MPEG-2. Moreover, a dynamic motion estimation technique is also proposed to further speed-up the decision process. Finally, we go a step further on our previous research efforts by combining the two aforementioned speed-up approaches. Our simulation results over more than 40 sequences at common intermediate format and quarter common intermediate format resolutions show that our proposal outperforms the MB mode selection of the rate-distortion optimization option of the H.264 encoder process by reducing the computational requirements by up to 90%, while maintaining the same coding efficiency. Finally, we conduct a comparative study of our approach with the most relevant fast inter-prediction methods for MPEG-2 to H.264 transcoder recently reported in the literature.

Index Terms—Data mining, H.264, inter prediction, MPEG-2, video transcoding.

I. INTRODUCTION

MPEG-2 [1] is nowadays the video compression standard most widely used, and the one into which a huge amount of money in infrastructure has been invested, e.g., DVD, cable and satellite television, video-on-demand services. Nevertheless, new and heterogeneous network technologies characterized by a wide range of transmission and error rates have entered the market: wireless local area networks, 3G cellular networks, and more. This has given origin to a new coding standard denominated H.264/advanced video coding

Manuscript received October 31, 2008; revised April 6, 2009 and July 28, 2009. First version published March 18, 2010; current version published May 5, 2010. This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04. It was also partly supported by The Council of Science and Technology of Castilla-La Mancha under Grants PEI09-0037-2328 and PII2109-0045-9916, and the University of Castilla-La Mancha under Project TC20091095. This paper was recommended by Associate Editor H. Sun.

G. Fernández-Escribano, J. L. Martínez, P. Cuenca, L. Orozco-Barbosa, and A. Garrido are with the Instituto de Investigación en Informática de Albacete, Universidad de Castilla-La Mancha, Albacete 02071, Spain (e-mail: gerardo@dsi.uclm.es; joseluismm@dsi.uclm.es; puenca@dsi.uclm.es; lorenzo@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.kalva@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.2010.2045914

(AVC) or MPEG-4 part 10 (May 2003) [2]. This new standard is capable, among other things, of maintaining the same quality of a sequence encoded using MPEG-2 while reducing the space necessary to store up the video material by half, and therefore, the network bandwidth for transmitting the encoded sequence. The research in the video transcoding field is particularly interesting due to the ample diffusion and use of MPEG-2 at the present time, and a clear interest by the industry on technologies to facilitate the migration from MPEG-2 to H.264 [3]. Nevertheless, the big differences existing between them make this task much more computational-intensive than in the case of other type of transcoders [4]. This letter presents an efficient MPEG-2 to H.264 Video Transcoder for being used with the H.264 baseline profile. The baseline profile is chosen for two main reasons: 1) it is the common profile used in most real-time applications, such as video, mobile TV, and video conference among others, and 2) it is one of the most common profiles used in the literature.

II. RELATED WORK

In the H.264 standard, the inter prediction is the most computational intensive task. The inter prediction is carried out in two steps: 1) motion estimation, and 2) motion compensation, also known as tree structured motion compensation. In order to evaluate the cost of the macroblock (MB) mode decision, the H.264 JM 14.0 reference software [5], implements various different evaluation methods, including the rate-distortion optimization (RDO), based on a Lagrange multiplier [6], among others. The interested reader may consult the work by Iain E.G. Richardson [7] for an in-depth description of the various methods used by the H.264 reference software.

A. Inter Prediction: Mode Decision

In H.264, inter frame motion estimation is performed using different MB sizes from 16×16 to 4×4 [7]. For each MB, all different sizes are evaluated and the one leading to the minimum rate-distortion (RD) cost is then selected. This “try all and select the best” philosophy guarantees the optimal block size for the final encoding but at the expense of a high-computational cost. Within the architecture of an MPEG-2 to H.264 transcoder, the inter prediction in this scenario needs to be enhanced. One of the most relevant mechanisms has been introduced by Chen *et al.* [8]. Furthermore, in our previous work [9], we identified that the mean and the variance

extracted from the post-processed residual MPEG-2 frames exhibit a high correlation with the H.264 MB mode decision. Moreover, we extended our analysis into other transcoder scenarios, e.g., H.263 to H.264 transcoder [10].

B. Inter Prediction: Motion Vector Refinement

Various mechanisms have been proposed to reduce the computational time needed for getting the motion vectors in a transcoding scenario [11], [12]. Closely related to the work herein, in one of our previous work [12], we have reduced the computational cost involved in the search window and the search range of the motion vectors extracted from the MPEG-2 bit stream.

III. MPEG-2 TO H.264 VIDEO TRANSCODER

In this letter, we focus our efforts on the Inter prediction process: the most computational intensive task involved in the transcoding process. The proposed approaches are based on the fact that MB coding mode decisions in H.264 video exhibit a high correlation with the distribution of the motion compensated residual in MPEG-2 video. data mining tools are used to exploit the existing correlation and derive decision trees for classifying the incoming MPEG-2 MBs into one of H.264 coding mode. The starting point of this process is simply based on a “look-and-feel” approach of the statistics applied to the MPEG-2 motion residual information. The major contributions reported in this letter and additions to our previous works [9], [10], [12] are the following.

- 1) In our previous works, we used multidecision trees. These kinds of trees are difficult to study since multiple choices are available for each decision node. In addition, they forced the standard to select only one MB with no options to evaluate other possibilities. In this letter, a new data mining process has been done (including new post-processed variables), in order to get a better classification. As a result, we replaced the previous trees with unbalanced binary trees allowing us to develop an H.264 encoder capable of a better selection of the MB mode.
- 2) The computational time can be kept low since a motion vector refinement is used with the MB mode decision approach. A reduced complexity motion estimation technique is proposed using the incoming MPEG-2 motion vectors to determine dynamically a motion vector refinement search window. In this way, the proposed approaches reduce the H.264 MB mode computation process into a decision tree lookup with very low complexity.
- 3) We have carried out a comparative study with the most prominent fast inter prediction methods for MPEG-2 to H.264 transcoding presented in the literature.

A. Data Mining: The Decision Tree

In this letter, we analyze the effect of incorporating nodes into the tree for selecting the best coding mode from a set of modes. In this way, by following the same line as in our

previous papers [9], [10] we further improve the transcoder operation by using a more suitable data mining process. New post processed variables, mean of variance and variance of the means, have been introduced in the data mining process, in order to develop a more generic tree. Second, binary trees have been used, where each leaf of the tree is a big bin including similar, in terms of correlation, MB, and sub-MB partition. Binary trees are more flexible allowing the development of an H.264 encoder capable of identifying the best MB mode decision in the bin. Since most of these ideas have been previously explained, we will only limit our study to the design and development challenges of the new kind of decision trees, as well as to their performance evaluation.

The JRip rule learner (Weka’s implementation of the RIPPER rule learner [14], proposed by William Cohen), a fast algorithm for learning “IF-THEN” rules, was used for creating the rules of the different nodes in the decision tree. RIPPER, like most classification learners, requires a set of examples to be represented as a vector of relatively simple features for learning and creating the rules.

The MPEG-2 video is decoded gathering the decoded MPEG-2 MB mode decision, the coded block pattern, and the MPEG-2 motion vectors. Moreover, additional operations here are needed for training, such as the mean and the variance of the 4×4 sub-blocks of the MPEG-2 motion residual MBs; the variance of the means, and the mean of the variances for each group of 4 means or 4 variances, are also used as part of the variables to be included for creating the tree. The class variable used for classifying the samples will be the decisions made by the H.264 reference software encoder, among the training sequence. Finally, in order to define the needed mode sets as needed, the H.264 reference software decisions will be modified for creating the binary decisions, i.e., 16×16 modes will create a set and the rest of the modes will be included into another one.

In our experiments, the training set has been built using 10 P-frames of the *Mobile Calendar* MPEG-2 sequence (VGA format), encoded at a higher bitrate than the typical broadcast encoding rates for the same quality (5 Mb/s). Since we have been interested in implementing a transcoder for the H.264 baseline profile B frames have not been included. Each P frame in the sequence, e.g., in the VGA format, consists of up to 1200 MBs. A large training set can thus be easily created using 10 P-frames from the video sequence (12 000 samples). The H.264 reference software decisions for the training set were obtained by encoding the MPEG-2 decoded sequence with a quantization parameter (QP) of 28 and RDO option enabled. Same parameters values and conditions will be used in the performance section for testing the model. After extensive experimentation, sequences containing regions varying from homogenous to high-detailed ones serve as good training sets (other good sample sequences are the *Flower Garden* and *Football* video sequences). It is worth to mention that since the *Mobile Calendar* sequence has been used for training the decision tree, this one has not been included in the tested sequence set. Note that for creating the set of rules for the different nodes the pruning option was set to False. Furthermore, the QP used by H.264 has been designed to

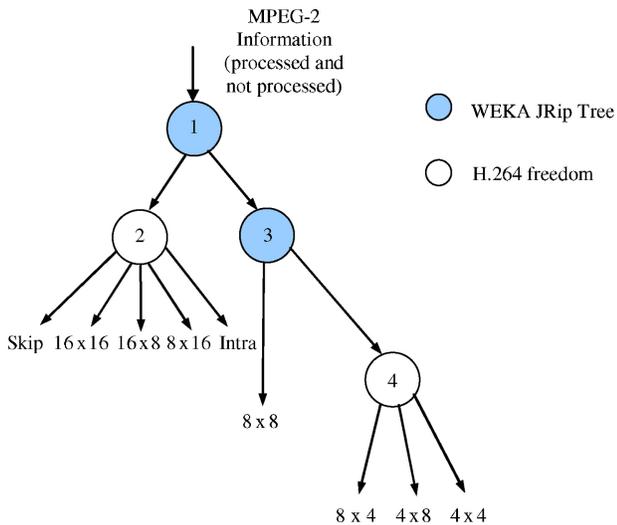


Fig. 1. Decision tree.

change the quantization step size and since the relationship between the QPs is well defined, this relationship can be used to adjust the mean and variance thresholds used in the Decision Tree [10]. We then presented an unbalanced binary decision tree (Fig. 1) based on a non-pruning JRip algorithm, with two kinds of nodes: blue nodes force to the encoder to follow a path in the encoding process, while the white ones are set free for choosing the one exhibiting the minimum cost over the RDO algorithm having been implemented in the reference software encoder.

B. Dynamic Motion Window

The motion estimation computation at the encoder depends largely on the search range used in the motion estimation process. A smaller search range reduces the computation efforts but it could reduce the video quality when achieving a sub-optimal prediction. Adapting the search range based on the motion vector of the incoming MB could reduce the motion estimation computation without severely affecting the peak signal-to-noise ratio (PSNR). To reduce the search range adaptively for every MB, an approach called dynamic motion window (DMW) based on the length of the MPEG-2 motion vectors has been developed, by exploiting the MPEG-2 motion vectors orientation. The approach presented in this letter for calculating them outperforms the previous mechanism proposed in one of our previous works [12]. The search range reduction depends on the MPEG-2 coding mode being used for a MB in the MPEG-2 encoder. The skip mode and Inter mode with zero motion vectors (no motion compensation coded) in MPEG-2 represent very simple MBs. In this way, the search range area is limited to 1 and 2 pixels respectively. On the other hand, the Intra MB mode in MPEG-2 can be encoded as Intra in the H.264 encoder or even as Inter with one of the many available MB mode partitions. Moreover, since the Intra mode in MPEG-2 does not give any insight on the motion, the search range is set to the maximum specified in the encoder configuration. In the case of MPEG-2 MB modes having motion vector information, the search range is

TABLE I
CHARACTERISTIC OF THE MPEG-2 INPUT SEQUENCES

Name	Resolution	Bitrate	GOP	MSR	Format
CIF	352 × 288	2 Mb/s	I11(P)	15	Progressive
QCIF	176 × 144	1 Mb/s	I11(P)	15	Progressive

MSR means motion search range.

determined by the area created by the circumference equation, centered in the (0, 0) point for each H.264 mode or sub-mode, with the length of the incoming MPEG-2 vector, such as, the radius of the circumference. In this way, the length of the MPEG-2 motion vector will be limited to the search range area.

IV. PERFORMANCE EVALUATION

The proposed MPEG-2 to H.264 Transcoder for the baseline profile has been implemented in the H.264/AVC JM 14.0 reference software [5]. It consists of two parts: an MPEG-2 decoder followed by an H.264 encoder. First, the transcoder fully decodes the MPEG-2 sequence, and the information required by the decision trees and the DMW mechanism is gathered during this stage. The H.264 MB mode decision is replaced by a simple mode assignment, or a range of them. Moreover, the MPEG-2 motion vectors are used for the DMW approach, and the low-complexity encoder performs the motion estimation just for the final MB mode(s) determined by the decision trees in the search range area fixed by the DMW mechanism, as previously explained.

A. Encoding Parameters

Simulations were conducted using different sequences encoded in different formats making a total of 44 videos; allowing us to evaluate the performance of the proposed algorithm when transcoding videos at commonly used resolutions: common intermediate format (CIF, 352 × 288) and quarter common intermediate format (QCIF, 176 × 144) for each sequence, if available. Table I shows the bitrates and the different parameters used for the encoded MPEG-2 video sequences used as input for the transcoder.

The parameters used in the H.264 encoder configuration file for testing the mechanism proposed against a cascade MPEG-2 to H.264 transcoder, were those included in the baseline profile of the JM14.0 reference software. Only four parameters have been changed in the configuration file. The H.264 video stream is transcoded using the same MPEG-2 group of pictures (GOP) structure, the number of reference frames is set to one and the *full search range* mode, for the motion estimation and compensated process was selected (value of 16 pixels for the Search Range, motion vector resolution of 1/4 pel). Finally, the RDO option, in the configuration file, has been enabled for all the simulations.

B. Metrics

The main metrics of interest are the results in terms of Δ Time, Δ PSNR and Δ Bitrate. These metrics were introduced in [9].

TABLE II
 Δ BITRATE, Δ PSNR AND TIME REDUCTION RESULTS FOR CIF AND QCIF SEQUENCES

Sequence, Number of Frames and QP s			Δ Bitrate, Δ PSNR and Time Reduction From the H.264 to MPEG-2 Reference Transcoder					
			CIF			QCIF		
			Time (%)	Δ PSNR (dB)	Δ Bitrate (%)	Time (%)	Δ PSNR (dB)	Δ Bitrate (%)
<i>Akiyo</i>	300	(28, 32, 36, 40)	-88.80	-0.014	0.40	-89.85	-0.024	0.57
<i>Coastguard</i>	300	(28, 32, 36, 40)	-92.47	-0.080	3.21	-92.33	-0.170	5.91
<i>Container</i>	300	(28, 32, 36, 40)	-91.99	-0.021	0.77	-91.23	-0.011	0.34
<i>Fast Food</i>	220	(28, 32, 36, 40)	-80.97	-0.134	5.83	-87.54	-0.242	7.53
<i>Flower Garden</i>	150	(28, 32, 36, 40)	-92.05	-0.064	3.43	-93.03	-0.111	3.56
<i>Football</i>	260	(28, 32, 36, 40)	-78.04	-0.113	5.48	-85.15	-0.243	9.05
<i>Foreman</i>	400	(28, 32, 36, 40)	-87.35	-0.076	3.49	-89.53	-0.099	3.42
<i>Harp</i>	220	(28, 32, 36, 40)	-90.66	-0.091	3.83	-91.51	-0.117	3.63
<i>Mother Daughter</i>	300	(28, 32, 36, 40)	-89.96	-0.029	1.03	-90.54	-0.026	0.82
<i>News</i>	300	(28, 32, 36, 40)	-89.46	-0.102	3.26	-91.51	-0.124	3.48
<i>Parade</i>	220	(28, 32, 36, 40)	-94.33	-0.044	2.31	-94.00	-0.054	1.64
<i>Paris</i>	1065	(28, 32, 36, 40)	-92.28	-0.102	3.44	-92.20	-0.134	3.68
<i>Salesman</i>	449	(28, 32, 36, 40)	-91.34	-0.032	1.27	-92.07	-0.054	1.85
<i>Sgi-ant</i>	220	(28, 32, 36, 40)	-90.84	-0.158	5.57	-91.36	-0.038	1.04
<i>Ship</i>	260	(28, 32, 36, 40)	-92.18	-0.012	0.38	-92.80	-0.007	0.19
<i>Silent</i>	300	(28, 32, 36, 40)	-89.98	-0.066	2.56	-91.04	-0.096	2.93
<i>Skyline</i>	260	(28, 32, 36, 40)	-89.24	-0.065	2.33	-91.18	-0.004	0.22
<i>Soft Football</i>	220	(28, 32, 36, 40)	-81.58	-0.080	5.24	-86.57	-0.192	7.07
<i>Stefan</i>	300	(28, 32, 36, 40)	-86.25	-0.151	7.05	-90.66	-0.222	7.05
<i>Suzie</i>	260	(28, 32, 36, 40)	-89.26	-0.047	1.74	-91.02	-0.075	2.52
<i>Tempete</i>	260	(28, 32, 36, 40)	-92.18	-0.060	2.88	-93.46	-0.097	3.11
<i>Waterfall</i>	260	(28, 32, 36, 40)	-92.77	-0.033	1.24	-93.30	-0.048	1.72
Average	6824	(28, 32, 36, 40)	-89.27	-0.072	3.03	-90.99	-0.099	3.24

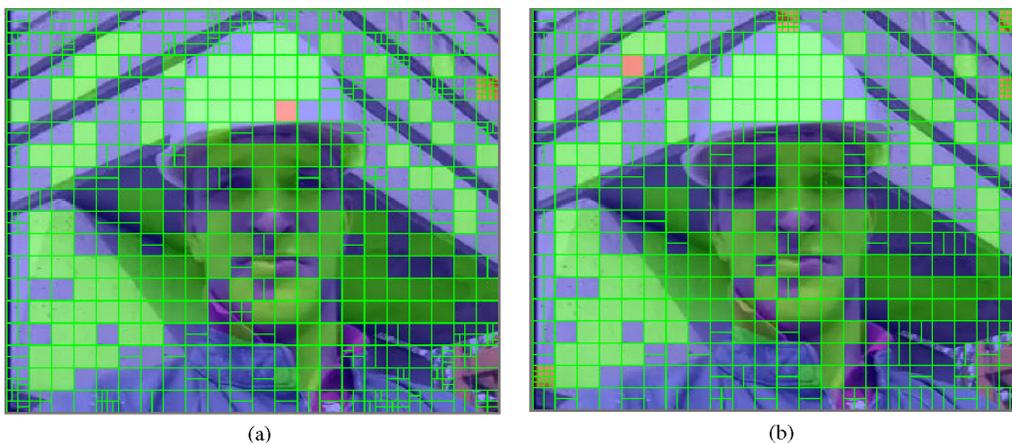


Fig. 2. MB mode decisions generated by (a) reference $QP = 28$, and (b) proposed transcoder $QP = 28$ for the first P-frame in the *Foreman* sequence.

C. Simulations Results

Table II shows the results in terms of Δ Time, Δ PSNR and Δ Bitrate. We also provide the average results for all the video sequences for each resolution. This should provide a good analysis on the performance of the proposed transcoder over all different kinds of video contents. Compared with the cascade MPEG-2 to H.264 reference transcoder, the proposed transcoder has a PSNR drop of 0.072 dB for a given bitrate, and a bitrate increase of 3.03% for a given PSNR in the CIF format, and 0.099 dB and a bitrate increase of 3.24% in the QCIF format; on average over all the tested video sequences. This negligible drop in RD performance is negligible taken

into account the computational savings which turn to be around 90% on average over all the tested video sequences.

Finally, Fig. 2 shows the H.264 MB mode decision obtained by using our proposed transcoder compared with the decision made by the cascade reference one, with a QP value of 28 for the *Foreman* sequence. The MB modes generated by the H.264 and the proposed algorithm are compared to measure the accuracy of the MB mode classification tree. The proposed algorithm exhibits very similar results to those obtained using the full estimation process in the H.264 encoder. The following color keys are used: red for Intra MB, green for skipped MB, and blue for Inter MB.

TABLE III
COMPARISON OF DIFFERENT FAST MB MODE DECISION ALGORITHMS. MPEG-2 TO H.264 TRANSCODING SCENARIO

Sequence	Format	Method	Δ Time (%)	Δ PSNR (dB)	Δ Bitrate (%)
Akiyo	CIF	Our Proposal	-88.80	-0.014	0.40
		Previous work [10]	-72.49	-0.052	1.78
Coastguard	CIF	Our Proposal	-90.54	-0.026	0.82
		Chen <i>et al.</i> [8]	-19.30	-0.140	2.30
Foreman	CIF	Our Proposal	-87.35	-0.076	3.49
		Previous work [9]	-81.95	-0.326	8.67
		Chen <i>et al.</i> [8]	-22.00	-0.130	3.00
Mother Daughter	CIF	Our Proposal	-89.96	-0.029	1.03
		Chen <i>et al.</i> [8]	-23.60	-0.120	4.50
News	CIF	Our Proposal	-89.46	-0.102	3.26
		Previous work [9]	-81.27	-0.302	7.33
		Chen <i>et al.</i> [8]	-23.90	-0.150	2.40
Paris	CIF	Our Proposal	-92.28	-0.102	3.44
		Previous work [10]	-73.63	-0.083	3.75
Silent	CIF	Our Proposal	-89.98	-0.066	2.56
		Chen <i>et al.</i> [8]	-24.00	-0.120	2.80
Stefan	CIF	Our Proposal	-86.25	-0.151	7.05
		Chen <i>et al.</i> [8]	-19.20	-0.070	2.30
Tempete	CIF	Our Proposal	-92.18	-0.060	2.88
		Previous work [10]	-71.01	-0.045	2.95

D. Comparison of Speeding-Up Mechanisms for the MPEG-2 to H.264 Video Transcoder

In this final section, we undertake a comparative analysis of our proposal to some of the most prominent MPEG-2 to H.264 transcoder algorithms [8]–[10]. Throughout our experiments, although the test conditions for all algorithms are not the same, a rather objective and fair comparison is still possible by following the Bjøntegaard and Sullivan’s common test rule [15]–[17]. The comparison metrics have been produced and tabulated based on the coding time (Δ Time), PSNR (Δ PSNR) and bitrate (Δ Bitrate) differences. It should be noted that we have only considered the sequences having been used in the works reported in the literature. As seen from Table III, our new proposal outperforms previously reported works including our own previous proposals [9], [10] in terms of Δ Time, Δ PSNR, and Δ Bitrate, getting results very close to the behavior of the cascade MPEG-2 to H.264 reference transcoder. As shown in Table III, the performance of our MPEG-2 to H.264 transcoder in terms of time saving, which is a critical issue in video transcoding applications, achieves the best results, with a negligible loss of video quality, and with a slight increment in bit rate with respect to the other methods in only a couple of sequences.

V. CONCLUSION

We have presented a low-complexity MPEG-2 to H.264 transcoder for the baseline profile, based on two speed-up approaches for the inter frame prediction in H.264, the most computational intensive task in an H.264 encoder. Data mining techniques jointly together with a motion vector refinement process can be used in order to define more flexible MBs mode sets. Moreover, our proposal outperformed all the algorithms previously reported for speeding up the various mechanisms of an MPEG-2 to H.264 transcoder. As a result, the transcoder

exhibits a very good performance over all of them, reducing the computational time by up to 90% on average.

REFERENCES

- [1] *Generic Coding of Moving Picture and Associated Audio*, ISO/IEC 13818-2, MPEG-2 Draft International Standard, 1994.
- [2] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264, May 2003.
- [3] T. Shanableh and M. Ghanbari, “Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats,” *IEEE Trans. Multimedia*, vol. 2, no. 2, pp. 101–110, Jun. 2000.
- [4] H. Kalva, “Issues in H.264/MPEG-2 video transcoding,” in *Proc. 1st IEEE Consumer Commun. Netw. Conf. (CCNC)*, Jan. 5–8, 2004, pp. 657–659.
- [5] *Reference Software to Committee Draft, JVT-F100 JM14.0*, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, 2008.
- [6] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, “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.
- [7] I. E. G. Richardson, “H.264/MPEG-4, part 10,” in *H.264 and MPEG-4 Video Compression*, Hoboken, NJ: Wiley, 2003, pp. 159–223.
- [8] G. Chen, P. R. China, Y.-D. Zhang, S.-X. Lin, and F. Dai, “Efficient block size selection for MPEG-2 to H.264 transcoding,” in *Proc. 12th Ann. Assoc. Comput. Machinery Int. Conf. Multimedia*, New York, Oct. 10–16, 2004, pp. 300–303.
- [9] G. Fernández-Escribano, H. Kalva, P. Cuenca, L. Orozco-Barbosa, and A. Garrido, “A fast MB mode decision algorithm for MPEG-2 to H.264 P-frame transcoding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 172–185, Feb. 2008.
- [10] G. Fernández-Escribano, J. Bialkowski, J. A. Gámez, H. Kalva, P. Cuenca, L. Orozco-Barbosa, and A. Kaup, “Low-complexity heterogeneous video transcoding using data mining,” *IEEE Trans. Multimedia*, vol. 10, no. 2, pp. 286–299, Feb. 2008.
- [11] Z. Zhou, S. Sun, S. Lei, and M.-T. Sun, “Motion information and coding mode reuse for MPEG-2 to H.264 transcoding,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, May 23–26, 2005, pp. 1230–1233.
- [12] G. Fernández-Escribano, H. Kalva, P. Cuenca, and L. Orozco-Barbosa, “Reducing motion estimation complexity in MPEG-2 to H.264 transcoding,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Beijing, China, Jul. 2–5, 2007, pp. 440–443.
- [13] I. H. Witten and E. Frank, “Implementations: Real machine learning schemes,” in *Data Mining: Practical Machine Learning Tools and*

- Techniques*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2005, pp. 187–283.
- [14] W. W. Cohen and Y. Singer, “A simple, fast, and effective rule learner,” in *Proc. 16th Natl. Conf. Artif. Intell.*, Orlando, FL, Jul. 18–22, 1999, pp. 335–342.
- [15] G. Bjøntegaard, “Calculation of average PSNR differences between RD-curves,” presented at the 13th VCEG-M33 Meeting, Austin, TX, Apr. 2001.
- [16] *Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material*, document VCEG-N81.doc, ITU-T VCEG, Sep. 2001.
- [17] *Evaluation Sheet for Motion Estimation*, Draft version 4, JVT Test Model Ad Hoc Group, Feb. 2003.