Model-Based Transrating of H.264 Coded Video

Naama Hait, *Member, IEEE,* and David Malah, *Life Fellow, IEEE*

Abstract—**This paper presents a model-based transrating (bit-rate reduction) system for H.264 coded video via requantization. In works related to previous standards, optimal requantization step sizes were obtained via Lagrangian optimization that minimizes the distortion subject to a rate constraint. Due to H.264 advanced coding features, the choices of quantization step size and coding modes are dependent and the rate control becomes computationally expensive. Therefore, optimal requantization algorithms developed for previous standards cannot be applied as they are. Hence, previous works on transrating in H.264 focused on changing the input coding decisions rather than on rate control, while requantization was addressed by a simple one-pass algorithm. Here we propose new model-based optimal requantization algorithms for transrating of H.264 coded video. The optimal requantization goal is to achieve the target bit rate with minimal effect on video quality. Incorporation of the proposed models serves two goals. For intra-coded frames, a novel closed-loop statistical estimator that overcomes spatial neighbors dependencies is developed. For inter-coded frames, the proposed macroblock-level models reduce the computational burden of the optimization. Overall, as compared to re-encoding (cascaded decoder–encoder), the proposed system reduces the computational complexity by a factor of about four, at an average PSNR loss of only 0.4 dB for transrating CIF/SIF sequences from** 2 **Mb/s to** 1 **Mb/s. In comparison with a simple one-pass requantization, the proposed algorithm achieves better performance (an average PSNR gain of 0.45 dB), at the cost of just twice the complexity.**

Index Terms—**Bit-rate control, H.264 video coder, requantization, transrating.**

I. Introduction

VIDEO SERVICES and multimedia applications use pre-encoded video in different formats for storage and transmission. As various user types require different formats and bit rates, a single copy of the encoded video cannot satisfy all users. One could store many copies of the video in the server, each encoded at a different format or bit rate, and send the bitstream that best matches the requirements of the user. However, such a server would suffer from very high storage costs and the chosen bitstream may not meet the exact user requirements. Therefore, servers typically store a single copy, pre-encoded at a high quality, and convert (or transcode) it on-line to match user-specific requirements. Transrating, which refers to bit-rate reduction within the same video format, can be achieved by a number of methods, such as frame rate reduction, spatial resolution reduction, and requantization of the transform coefficients. In this paper, we examine model-based transrating via requantization of the transform coefficients, for the state-of-the-art H.264 video coder.

Optimal requantization for MPEG-2 encoded video was suggested in [1] by minimizing the frame's distortion subject to its target bit rate. In that work, the optimization procedure became an expensive exhaustive search since it evaluated the rates and the distortions for each picture region (e.g., a macroblock) at multiple requantization steps exhaustively, with no models. Previous works that did use analytic models for optimal bit allocation [2], [3] aimed at encoding the *original input* video, using earlier video coding standards.

H.264 is currently the state-of-the-art video coding standard. Its advanced coding features offer an improvement in the coding efficiency by a factor of about two over MPEG-2 [4] at the expense of higher complexity. As the choices of quantization step size and coding modes are dependent, the rate control becomes computationally expensive. Therefore, previous works on transrating in H.264 [5]–[7] focus on changing the input coding decisions (intra-prediction modes and motion) rather than on the rate control, and requantization is addressed by a simple one-pass algorithm [5].

In this paper, new model-based optimal requantization algorithms for transrating of H.264 coded video are developed and examined. The models incorporated in this paper relate the rate and the distortion to the fraction of zeroed quantized transform coefficients ρ [8] rather than to the step size itself. At first, frame-level bit allocation is determined by minimizing the overall distortion over a group of frames, such that the target average bit rate is achieved. To keep a smooth constant video quality, the frame distortions are equalized. Then, intra- and inter-frames are re-quantized, separately.

For *intra-coded* frames, the spatial prediction in H.264 introduces dependencies between neighboring residual blocks. As a result, the residual coefficients to be requantized are not available when needed for requantization step-size selection. Therefore, the estimation of the relation between ρ and the requantization step size becomes a challenging task. To this end, we propose a novel closed-loop statistical estimator, which outperforms the simple open-loop estimator.

For *inter-coded* frames, we propose to solve an optimal nonuniform requantization problem. The requantization step size for each macroblock is chosen such that the overall frame distortion is minimized subject to a rate constraint and a limitation of the change in the requantization step size in consecutive macroblocks that helps to improve the subjective quality. To solve that regularized optimization problem, we suggest to

Manuscript received March 3, 2008; revised October 29, 2008. First version published April 7, 2009; current version published August 14, 2009. This paper was supported in part by STRIMM consortium-MAGNET program of the Ministry of Trade and Industry via Samuel Neaman Institute. This paper was recommended by Associate Editor J.-F. Yang.

N. Hait is with Elbit Systems Ltd., Advanced Technology Center, Haifa 31053, Israel (e-mail: naama.hait@gmail.com).

D. Malah is with the Electrical Engineering Department, Technion-Israel Institute of Technology, Haifa 32000, Israel (e-mail: malah@ee.technion.ac.il).

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.2009.2020334

1051-8215/$26.00 © 2009 IEEE

extend the Lagrangian optimization (see [1]) by an inner loop that applies dynamic programming. To reduce the computational burden of the optimization, we use rate-distortion models at the macroblock level. As the models suggested in the literature are not suitable for macroblock level coding in H.264, we developed macroblock level rate-distortion models adapted to H.264 requantization. Since the recommended software encoder [9] eliminates very sparse blocks, we also examine the option of extending the optimal requantization by selective coefficient elimination. In addition, we incorporated some human visual system (HVS)-based considerations in the system design to gain a higher perceptual quality, as a secondary focus of the paper. Partial details and preliminary results were reported in [10], [11] dealing with transrating of intra-coded and inter-coded frames, respectively. This paper describes in full the complete proposed transrating system, including the final algorithms and overall system performance evaluation.

Section I-A provides a short overview of existing ρ-domain models. Section I-B discusses the chosen transrating architectures for intra-coded frames and inter-coded frames. We assume that the reader is familiar with the basics of the H.264 coder [4], [12].

A. ρ-Domain Rate-Distortion Models

Different models in the literature suggest different relations for rate versus quantization step size. In [8], [13], the ρ-domain source model is suggested, where ρ is the fraction of zero coefficients among the quantized transformed coefficients in a frame. The model assumes that there is a strong linear relation between ρ and the actual frame's bit rate: coarser quantization step sizes generate more zero coefficients (and hence increase ρ) while decreasing the rate (where the rate here refers to the bits spent on coding the transform coefficients). Therefore, the suggested *rate versus* ρ relation is [8], [13]

$$R(\rho) = \theta \cdot (1 - \rho) \tag{1}$$

where R is the rate and θ is a parameter determining the slope. According to this equation, for $\rho = 1$ all the quantized coefficients are zeroed and thus the coding rate should approach zero. It is also argued in [8], [13] that the rate–ρ model is more robust than a rate–quantization-step model: the observed rate–ρ curves for both I- and P-frames share a very similar pattern, whereas the rate–quantization step-size curves change between different frame types.

The ρ-domain is also more suitable for modeling the distortion than the quantization step-size domain as it is defined within a finite range $0 \leq \rho \leq 1$ and follow a more robust and regular behavior. In [3], an exponential-linear model for the MSE distortion in the ρ-domain was suggested as

$$D(\rho) = \sigma^2 \cdot e^{-\alpha \cdot (1-\rho)} \tag{2}$$

where σ^2 is the variance of the transformed coefficients and $\alpha > 0$ is a model parameter. As $\rho \to 1$ and all the quantized coefficients are zeroed, the distortion approaches the σ^2 bound.

These models were derived for describing the rate and the distortion at the frame level, and were found quite accurate in [3], [8] and [13], when tested for standards such as MPEG-2 and H.263, and were also used in [14], [15] for H.264. However, we found that for H.264 requantization at the macroblock level, these models are not good descriptors of the empirical data. Therefore, in Section IV-B, we suggest different ρ-domain models, specifically adapted for H.264 requantization.

B. Architectures for Transrating of Coded Video

In this section we outline four transrating architectures that provide different compromises between quality and computational complexity. Re-encoding is a naive and straightforward architecture [16], [17], where a decoder and encoder are cascaded. It has the highest computational complexity among transrating architectures, as it makes new coding decisions, which also involve performing motion estimation (ME).

The open-loop transrater has the lowest computational complexity for requantization-based transrating [16]–[19], but is subject to a drift error that degrades the video's quality [17], [19]. The residual's transform coefficients are dequantized and then requantized at a coarser step size. Expensive operations such as ME and transforms are avoided and there is no need for a frame-store.

The *full decoder-guided encoder* (FD-GE) architecture [16], [17], [20] reduces the computational complexity as compared to re-encoding, without introducing a drift error, by reusing the input coding decisions (e.g., motion vectors and intra-prediction modes) during the new residual encoding.

The *partial decoder–partial encoder* (PD-PE) architecture [16], [17], [19]–[21] further simplifies the FD-GE by reconstructing just the residual signal in the pixel domain, rather than the fully decoded picture. It performs a closed-loop correction to compensate for the drift error by applying the motion compensation (MC) once (in the joint transrater loop) instead of twice (during both decoding and encoding).

Transrating in H.264 requires distinguishing between intra-coded frames and inter-coded frames. The spatial prediction in intra-frames use previously decoded neighbor pixels in the same frame to predict the current block pixels. Therefore, any mismatch between the transcoder and the encoder/decoder introduces a drift error that propagates throughout the frame [22]. Since some of the operations are not linear (due to rounding and clipping), this drift cannot be fully compensated. Therefore, to avoid the drift error, intra-frames should be fully decoded into images in the pixel domain and then encoded [22] using the FD-GE architecture. The *guided encoding* allows either reuse of the input intra prediction modes or their selective modification, as will be discussed in Section III-B. The selection of the requantization step size for intra-frames is discussed in Section III-A.

Inter-coded frames are transrated using the PD-PE architecture, as the temporal drift error is very small and it takes a number of frames before the accumulated error is noticeable. The reason is that the MC is approximately linear up to rounding and clipping operations, which can be neglected.

H.264 defines an in-loop deblocking filter, which may be applied on the fully decoded pictures in the pixel domain. We assume that the filter is disabled, so the pictures need not be fully decoded and the PD-PE architecture can be applied [22].

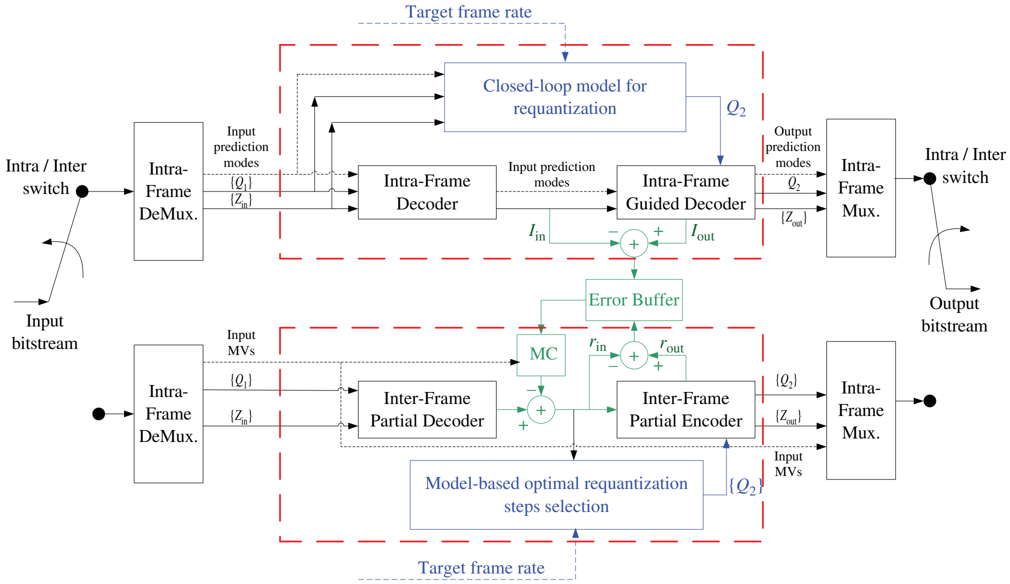

Fig. 1. Block diagram of the proposed transrating system. For each frame, the input bitstream is first parsed to read the input quantized coefficients indices $\{Z_{\text{in}}\}$, the input quantization steps $\{Q_1\}$, and the input prediction modes/motion vectors (MVs). Intra-coded frames are transrated using a FD-GE architecture (top block enclosed in a red dashed line). The guided encoder outputs are the output quantized coefficients indices $\{Z_{\text{out}}\}$, the requantization step Q_2, and the output intra prediction modes, all of which are entropy-encoded and written in the output bitstream. The requantization step Q_2 is found using the closed-loop model for requantization. The transrating error is saved in the error buffer, as part of a closed-loop correction scheme. Inter-coded frames are transrated using a PD-PE architecture (bottom block enclosed in a red dashed line). The partial decoder reconstructs the residual in the pixel domain, and then performs a closed-loop compensation, to account for the transrating errors introduced in the previous frames (the difference between r_{in} and r_{out}). The corrected residual r_{in} is fed into the model-based optimal requantization steps selection algorithm to find the optimal requantization steps $\{Q_2\}$. The corrected residual r_{in} is subtracted from the transrated residual r_{out} to form the transrating error, and saved in the error buffer.

Still, in Section V we propose modifying our transrating system to support the transrating of an input sequence for which the deblocking filter was enabled.

Intra-coded blocks inside inter-frames are transrated using the PD-PE architecture too (with the appropriate changes, e.g., the MC block is replaced by the spatial predictor, etc.) though this is not the recommended architecture for them. Therefore, transrating inter-frames with many intra-coded blocks using PD-PE architecture do cause some drift, but these cases are rather infrequent. The rate control algorithm handles these blocks as if they were inter-coded blocks. A block diagram of the proposed transrating system is depicted in Fig. 1.

The paper is organized as follows. Section II describes using ρ-domain rate-distortion models for bit allocation among transrated video frames in a group of pictures (GOP). The algorithm for transrating intra-coded frames is described in Section III, where the main mean for bit-rate reduction is model-based uniform requantization (in Section III-A) and a secondary mean is modification of the prediction modes (in Section III-B). The algorithm for transrating inter-coded frames is presented in Section IV, using model-based optimal nonuniform requantization. The optimization algorithm is described in Section IV-A, and new macroblock-level models in Section IV-B. Section V summarizes the main simulation results and Section VI concludes the paper.

II. Model-Based Optimal GOP-Level Bit Allocation

To achieve the bit-rate reduction, we apply rate control algorithms at two levels. The coarser level determines the bit allocation to frames in a GOP, which is discussed in this section. The finer level allocates the bits to each frame encoding units (e.g., macroblocks) to achieve the frame target rate, which will be discussed in Sections III-A and IV-A for intra- and inter-frames, respectively.

The encoded bitstream describes two types of data. The “texture bits” describe coding the quantized residual transform coefficients, whereas the “overhead bits” describe the coding modes, MB types, etc. When the input coding modes are reused, most of the overhead bit count remains. Therefore, we assume that the change in the overhead bits due to transrating is negligible. To reduce the bit rate at an average transrating factor $BRfactor$, one could reduce each frame’s bit rate by the $BRfactor$ factor. But, in H.264 the overhead bits are not negligible and such a simple frame-level bit allocation may leave too few texture bits for coding the residual.

Thus, we would like to find the optimal texture bits allocation to the frames of that GOP; that is, to minimize the overall GOP distortion subject to the average rate constraint. This optimization problem was solved in [3] analytically by using the ρ-domain rate-distortion models. Subjectively, the overall sequence distortion is more tolerable when all frames suffer similar distortion [2], [23], [24]. In [2], [24], each frame’s target distortion was set as the average distortion of the previously encoded frames, and then its target rate was extracted using the ρ-domain rate-distortion models. Therefore, these works do not allocate the texture bits optimally. In [25], a new optimal bit allocation problem was analytically solved for each encoded frame. For each frame, the target bit rate was calculated such that all the remaining frames in the GOP would have an equal distortion subject to the rate constraint, using a modified distortion model in the ρ-domain.

We propose to analytically solve a single optimal bit allocation problem per GOP, prior to its transrating (assuming that a GOP delay is tolerable). We minimize and equalize the transrating distortion over all the frames of that GOP, and the optimization problem formulation becomes

$$\min_{\{R_k\}} \sum_{k=1}^{N} D_k(\rho_k), \quad \text{subject to:} \tag{3}$$

$$\sum_{k=1}^{N} R_k(\rho_k) \le R_{GOP,\text{target}}$$

$$D_1(\rho_1) = D_2(\rho_2) = \cdots = D_N(\rho_N)$$

where N is the number of frames in the GOP, R_k and D_k are the rate and the distortion of frame #k where $1 \le k \le N$ and $R_{GOP,\text{target}}$ is the target rate for the N frames together. We use the ρ-domain models (1) and (2) to obtain an analytic solution (using Lagrangian parameters to convert the constrained problem into an unconstrained problem)

$$R_k = \xi_k \cdot \left[\ln(\sigma_k^2) - \frac{\sum_{l=1}^{N} \xi_l \cdot \ln(\sigma_l^2) - R_{GOP,\text{target}}}{\sum_{l=1}^{N} \xi_l} \right] \tag{4}$$

$$D_k = \exp\left(\frac{\sum_{l=1}^{N} \xi_l \cdot \ln(\sigma_l^2) - R_{GOP,\text{target}}}{\sum_{l=1}^{N} \xi_l} \right) = \frac{1}{N} \sum_{k=1}^{N} D_k \tag{5}$$

where the resulting D_k is a constant and $\xi_k = (\theta_k/\alpha_k)$. This solution allocates more texture bits for the intra-coded frame (as compared to the allocation that does not pose the equal distortion constraint) to keep an equal distortion over all the frames. The model parameters are adaptively extracted from the coded input for each frame. At the end of each frame's encoding, the deficit or surplus is uniformly distributed among the remaining frames in the GOP.

III. Intra-Frames Transrating

The spatial prediction introduced in intra-coded frames require a full decoding and guided encoding architecture (FD-GE) in order to avoid a drift error (see Section I-B). The main mean for bit-rate reduction is via transform coefficients requantization (discussed in Section III-A). A secondary mean is via modification of the prediction modes, to increase the coding efficiency (discussed in Section III-B).

A. Model-Based Uniform Requantization

For intra-coded frames, we propose using uniform requantization for two reasons. One is that the typical bit budget for intra-frames is sufficiently high (as compared to inter-frames) to allow a frame-level rate control. The other reason is that the spatial prediction introduces block dependencies that greatly increase the computational complexity and memory requirements of solving an optimal nonuniform requantization problem. Due to these dependencies, the residual coefficients to be requantized are not available when needed for the requantization step-size selection. The uniform requantization step size is found using two ρ-domain models: the linear rate-ρ model and a new $\rho - Q_2$ model, where Q_2 is the requantization step size. The evaluation of the linear rate-ρ model is fairly simple and is described in Section III-A.1. Most of the effort is aimed at estimating the $\rho - Q_2$ model. Section III-A.1 reviews the open-loop approach for evaluating the $\rho - Q_2$ relation and explains its shortcomings. Section III-A.2 proposes a closed-loop statistical estimator for the $\rho - Q_2$ relation. It overcomes the block dependency problem by modeling the correction signal of the requantizated residual.

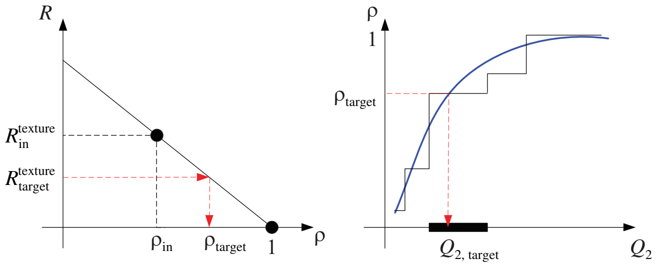

Fig. 2. Uniform requantization using a rate–ρ model. Left: rate–ρ relation, the dark circles are at $(\rho_{\text{in}}, R_{\text{in}}^{\text{texture}})$ and $(1, 0)$, from which θ is estimated. Right: $\rho(Q_2)$ relation, blue smooth curve: closed-loop estimator, black staircase curve: open-loop estimator. Given $R_{\text{target}}^{\text{texture}}$, we extract ρ_{target} and then find the corresponding $Q_{2,\text{target}}$ using the closed-loop $\rho(Q_2)$ estimator. Using the open-loop $\rho(Q_2)$ estimator, there is an uncertainty interval regarding $Q_{2,\text{target}}$ choice, as illustrated by the thick black line.

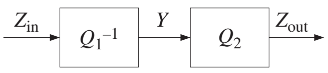

Fig. 3. Open-loop requantization scheme.

1) Open-Loop Approach for Requantization Step-Size Selection: We use the linear rate–ρ model (1) to set a uniform requantization step size for an I-frame. The model parameter θ is estimated using the input rate–ρ point, $(\rho_{\text{in}}, R_{\text{in}}^{\text{texture}})$ and an anchor point at $(1, 0)$, as depicted on the left of Fig. 2. Given the target rate for that frame $R_{\text{target}}^{\text{texture}}$ we extract the expected fraction of zeros by (6). The next step is to estimate the relation between ρ and the requantization step size Q_2 as a $\rho = f(Q_2)$ lookup table, to be discussed in Section III-A.2. Then, the target step is found by (7)

$$\rho_{\text{target}} = 1 - R_{\text{target}}^{\text{texture}}/\theta \tag{6}$$

$$Q_{2,\text{target}} = f^{-1}(\rho_{\text{target}}). \tag{7}$$

Due to spatial prediction, requantization of the residual at one block changes the residual in neighboring casual blocks. To avoid a drift error, intra-frames are fully decoded into pictures in the pixel domain, and then encoded. But, estimating the $\rho(Q_2)$ relation this way requires multiple encoding of the picture at different Q_2 steps, which is not practical.

The simplest $\rho(Q_2)$ estimator is the *open-loop estimator*, evaluated from the output of the scheme depicted in Fig. 3. The input quantized indices Z_{in} are dequantized using the input quantization step size Q_1 to yield the residual transform coefficients Y. When Y is requantized using a quantizer with step size Q_2 and deadzone Δz, the output indices are $Z_{\text{out}} = sign(Y) \cdot \lfloor (|Y|/Q_2) + \Delta z \rfloor$. Therefore, all transform coefficients that fall in the interval $[-t(Q_2), t(Q_2)]$ are requantized to zero, where $t(Q_2) = (1 - \Delta z)Q_2$. For intra-frame, $\Delta z = 1/3$

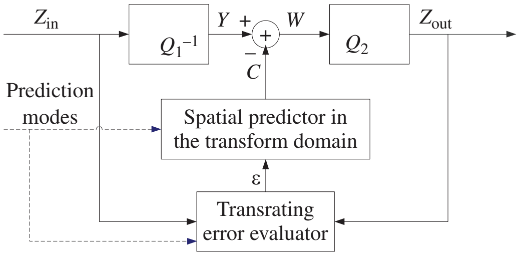

Fig. 4. Closed-loop modeling scheme for estimating $\rho(Q_2)$. The transrating error ε is fed into the predictor to yield the correction signal C. Then, $\rho(Q_2)$ is estimated based on $W \triangleq Y - C$.

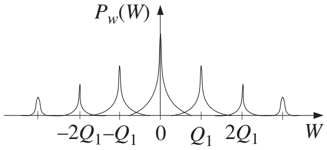

Fig. 5. Schematic illustration of the probability distribution of W.

and therefore $t(Q_2) = (2/3)Q_2$. This process is repeated for each Q_2 step size, to derive the $\rho(Q_2)$ relation.

This open-loop $\rho(Q_2)$ estimator cannot track the changes in the residual and therefore, it has two disadvantages. One is that it is not accurate enough at moderate to coarse requantization, where large changes in residual intensity cause a large drift error. The other is its staircase characteristic, (see staircase curve on the right of Fig. 2). Given a target ρ value, the estimator may encounter an uncertainty as to which requantization step size to choose. The uncertainty interval is illustrated by the thick black line on the right of Fig. 2.

2) Closed-Loop Estimation of $\rho(Q_2)$: Since the residual coefficients to be requantized are not available in advance of setting Q_2, the estimation of $\rho(Q_2)$ is not trivial. To estimate it more accurately than the open-loop approach, we propose [11] to model the process that the input coefficients Y undergo to become the residual coefficients to be requantized. To this end, we need not estimate the value of every single coefficient, but rather their statistical distribution. We first describe the model's scheme and then provide a statistical description of the residual coefficients to be requantized.

a) Closed-loop residual modeling architecture: We propose to estimate $\rho(Q_2)$ using a model that is based on a closed-loop residual architecture in the transform domain, as depicted in Fig. 4. The closed-loop estimator statistically models the required correction of the requantized residual coefficients, thereby overcoming the dependency problem. The scheme in Fig. 4 is merely used in order to model the distribution of residual coefficients to be requantized, from which ρ is estimated. During actual transrating, we fully decode the picture, estimate the $\rho(Q_2)$ relation using this model, estimate the linear rate$-\rho$ model (as described in Section III-A.1), choose Q_2 that meets the target rate (as illustrated in Fig. 2), and then encode the picture once (by performing spatial prediction, transforming the obtained residual, and requantizing) using the chosen Q_2.

Instead of evaluating $\rho(Q_2)$ based on Y, the closed-loop $\rho(Q_2)$ estimator evaluates how many of the corrected transform coefficients W (see Fig. 4) fall in the deadzone interval. The corrected residual is defined as $W \triangleq Y - C$, where C is the correction signal in the transform domain. This signal is formed by feeding the transform domain transrating error ε into the transform domain spatial predictor (performs the equivalent operation to spatial prediction in the transform domain [26]). Due to some nonlinearities (rounding and clipping operations), the transrating error ε cannot be defined simply as the requantization error. Rather, it is defined as the transform of the difference between the decoded output and input images, where the output image is decoded using the requantized indices $Z_{\text{out}} = Q_2(W)$.

In order to evaluate $\rho(Q_2)$ from W, we first characterize the statistical distributions of Y and C, and then find how W is distributed. Since the input transform coefficients Y have values that are multiples of the input quantization step size Q_1, their distribution is discrete: $p_Y(y) = \sum_{l=-L}^{L} p_l \cdot \delta(y - lQ_1)$ where $\delta(y)$ is the unit impulse function, L is the smallest integer such that $|Y| \leq LQ_1$, and $\{p_l\}_{l=-L}^{L}$ are extracted from the input coefficients.

The correction signal C is modeled as a continuous distribution. Since this signal cannot be explicitly extracted from the input stream, most of the effort is aimed at its characterization and its statistical modeling. Once the distribution of C is obtained, the next step is to find the distribution of $W = Y - C = Y + (-C)$. A schematic illustration of the distribution of W is depicted in Fig. 5. Since we cannot assume that C is independent of Y, we use the joint probability of $(Y, -C)$: $p_{Y,-C}(y, c) = p_{-C|Y}(c|y) \cdot p_Y(y)$ to calculate the cumulative distribution of W in (8)

$$\begin{aligned} Pr.(W \leq w_0) &= \int_{-\infty}^{\infty} \int_{-\infty}^{w_0 - y} p_{Y,-C}(y, c)\,dc\,dy \qquad (8) \\ &= \sum_{l=-L}^{L} p_l \cdot \int_{-\infty}^{w_0 - lQ_1} p_{-C|Y}(c|Y = lQ_1)\,dc. \end{aligned}$$

Therefore, the closed-loop $\rho(Q_2)$ evaluation is given by

$$\rho(Q_2) = Pr.(|W| \leq t(Q_2)) = \sum_{l=-L}^{L} p_l \cdot \phi(l|Y) \qquad (9)$$

where $\phi(l|Y) = \int_{-t(Q_2)-lQ_1}^{t(Q_2)-lQ_1} p_{-C|Y}(c|Y = lQ_1)dc$.

Lacking a known model for the correlation between Y and C, we are left with the unfeasible task of modeling $\phi(l|Y)$, for every possible value of Y (corresponding to $|l| \leq L$). From observations, we found that a reasonable approximation is to distinguish between zero and nonzero inputs: that is, to model $\phi(0|Y = 0)$ and $\phi(l|Y \neq 0)$ separately. In that case, the model in (10) for $\rho(Q_2)$ is simpler, as there are two possible input dependencies instead of $2L + 1$. To complete the evaluation of $\rho(Q_2)$, we now address the evaluation of $\phi(0|Y = 0)$ and $\phi(l|Y \neq 0)$, by characterizing the correction signal C and modeling its distribution

$$\rho(Q_2) = p_0 \cdot \phi(0|Y = 0) + \sum_{l=-L, l\neq 0}^{L} p_l \cdot \phi(l|Y \neq 0). \qquad (10)$$

b) Correction signal characterization: To ease its statistical modeling, the correction signal C is partitioned into homogenous *data groups* that share the same characteristics, according to three partitioning criteria. The first partition of the data is according to its spatial prediction modes that spectrally shape the white error ε. The second partition distinguishes the affected coefficients from the unaffected coefficients. Affected coefficients are those coefficients that are changed as a result of spatial prediction, whereas unaffected coefficients have a zero correction signal. This coefficient partition is predefined for each prediction mode, e.g., the dc prediction affects just one transform coefficient out of a 4 × 4 integer cosine transform block. The advantage of the affected/unaffected coefficients classification is that the $\rho(Q_2)$ relation for the unaffected coefficients can be evaluated as in the simple case of an open-loop estimator, thereby reducing the complexity of evaluating the $\rho(Q_2)$ relation. The third partition distinguishes between the corrections applied to zero/nonzero input coefficients. Next, a probability distribution is fitted to each data group allowing evaluation of its $\rho(Q_2)$ relation according to (10).

c) Correction signal modeling using a Γ distribution: To evaluate (10) for each data group, a statistical description of $\phi(0|Y = 0)$ and $\phi(l|Y \neq 0)$ is required. To study this issue, we evaluated the correction signal C offline, according to the scheme of Fig. 4, and performed the partitioning into data groups. We then found that the Γ distribution is a good descriptor of each of the correction signal partitions. The probability density function for the two-sided Γ distribution [27] is defined as $p_X(x;\beta) = [1/(2\sqrt{\pi})]\sqrt{(\beta/|x|)} \cdot \exp\{-\beta|x|\}$, where $\beta > 0$ is a scale parameter, whose decrease results in a wider distribution. The Γ cumulative distribution function is defined by (11), where $\Gamma(a, 0.5) \triangleq \int_0^a t^{-0.5}\exp(-t)dt$

$$Pr.(X \leq x; \beta) = \frac{1}{2} + sgn(x)\frac{1}{2\sqrt{\pi}}\Gamma(\beta|x|, 0.5). \qquad (11)$$

To complete the offline model evaluation, an ML estimator was applied to find the scale parameter β for the affected correction coefficients, for each prediction mode, while distinguishing $\beta^{C|Y=0}$ from $\beta^{C|Y\neq0}$ for the zero/nonzero input coefficients, respectively. Using (11) and these estimated parameters, the functions $\phi(0|Y = 0)$ and $\phi(l|Y \neq 0)$ take the form of (12), and $\rho(Q_2)$ can be evaluated for each data-group by substituting (12) into (10). Then, all data-group $\rho(Q_2)$ relations are linearly weighted (according to their size) to obtain the frame-level relation

$$\begin{aligned}\phi(0|Y = 0) &= Pr.(|C| \leq t(Q_2); \beta^{C|Y=0}) \qquad (12)\\ \phi(l|Y \neq 0) &= Pr.(|C + lQ_1| \leq t(Q_2); \beta^{C|Y\neq0}).\end{aligned}$$

As stated earlier, in a real-time scenario, the scheme of Fig. 4 is not implemented. Therefore, the correction signal C is not available and the ML estimator for β cannot be used. Observations show that the value of β monotonically decreases with Q_2, as coarser requantization generates a transrating error ε with a wider dynamic range, which in turn generates a correction signal with a wider dynamic range when fed back to the predictor. However, the great variability in the $\beta - Q_2$ relation over different data-groups complicates its modeling. Therefore, we suggest decomposing this relation into two separate models: β versus $||\varepsilon||_1$ and $||\varepsilon||_1$ versus Q_2, as illustrated in Fig. 6. The β versus $||\varepsilon||_1$ relation is modeled by $\beta = \beta_0/||\varepsilon||_1$. When the transrating error is zero, a correction signal is not generated, and hence $\beta \rightarrow \infty$. The $||\varepsilon||_1$ versus Q_2 relation was empirically fitted using the monotonically increasing function $||\varepsilon||_1 = a_1 \cdot (ln(Q_2))^2 + a_2$, whose parameters a_1, a_2 are functions of the input "initial conditions" Q_1 and $||Y||_2$.

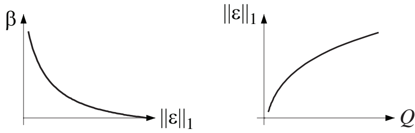

Fig. 6. Decomposition of the β versus Q_2 relation, using $||\varepsilon||_1$.

To summarize, the modeling steps are as follows.

1) Segment the transform coefficients into data groups (according to the prediction modes, affected/unaffected coefficients, and zero/nonzero input coefficients).
2) For each data group, evaluate the β distribution parameter from the input data in two stages.
   a) Model the $||\varepsilon||_1$ versus Q_2 relation (fit a_1, a_2).
   b) Model the β versus $||\varepsilon||_1$ relation (fit β_0).

   Substitute (12) into (10) to evaluate its $\rho(Q_2)$ relation.
3) Linearly weight the obtained $\rho(Q_2)$ relations for the different data parts according to their relative size to get the frame level $\rho(Q_2)$ relation.

If the input frame is not uniformly quantized during the first encoding, an additional data partition according to the initial quantization step is added to the data groups segmentation. Section V-B.1 compares the $\rho(Q_2)$ evaluation using the proposed model to the true data and the open-loop estimator.

B. Modification of Prediction Modes

The proposed architecture used for transrating intra-coded frames (see Section I-B) requires full decoding and encoding in order to avoid a drift error. Although we have to fully decode the frame, we need *not* fully encode it by means of a computationally expensive full prediction modes search. Rather, we perform a *guided encoding*, which uses already encoded information from the input bitstream. One option is to reuse the input prediction modes. The other option is to selectively modify the input prediction modes where the coding efficiency is expected to improve.

Spatial prediction in intra-coded frames significantly increases the coding efficiency when the coding modes are appropriately selected. The H.264 encoder chooses the best mode m_i^* by (13), where d_i and r_i are the distortion and the number of bits spent for block i, QP is the quantization parameter, and λ is the Lagrangian parameter [9]: $\lambda(QP) = 0.85 \cdot 2^{[(QP-12)/3]}$

$$m_i^* = \arg\min_m\{d_i(m, QP) + \lambda(QP) \cdot r_i(m, QP)\}. \qquad (13)$$

As the bit rate is reduced, the quality is degraded and fine details are less likely to be preserved. The observed trend regarding the encoder's intra coding decisions shows that as the bit rate is reduced, larger prediction blocks are chosen (more 16 × 16 partitions) and the frequency of "simple" modes (horizontal, vertical, and dc prediction) increases at the expense of the more complex "diagonal" modes for the remaining 4 × 4 partitions. However, for some blocks, "complex" modes usage significantly improves the coding efficiency, so these modes cannot be completely discarded from the search.

A previous work [28] considered modifying prediction modes originally coded as 4×4, as most of the coding gain is expected due to their modification. That work used the number of bits spent on coding the original MB as a prior to discern the smooth from the highly detailed MBs. Smooth MBs were examined for 16×16 prediction, whereas highly detailed MBs were examined for 4 × 4 predictions. The decision whether to change the mode, in that work, was based solely on the distortion. Such an approach may yield large rate deviations, as the best mode selection is correlated with its rate-distortion cost at the current bit rate working point.

We suggest choosing the best new modes while considering both the input prior and the HVS characteristics. The input bit consumption is used as the input prior and the distortion is weighted according to the HVS characteristics. Our best mode choice is given by

$$m_i^* = \arg\min_{m \in M} \{d_i(m, QP) + \lambda(QP) f_i^{\text{HVS}} r_i(m, QP)\} \quad (14)$$

where d_i is the requantization distortion, M is the subset of modes found using the input prior, and f_i^{HVS} is the perceptual weight given to block i, as we explain next.

1) Input Prior: We suggest to use the input prediction mode to narrow down the number of searched modes. For MBs initially encoded at a 16 × 16 prediction and for the chrominance components, the input mode is reused so no new modes are searched for. For MBs initially encoded as 4 × 4 we determine the subset M of modes that are searched for by classifying the picture macroblocks into three groups according to their input bit consumption:

a) G^L group (the lowest 30% input bits consumption). Blocks are assumed to be relatively smooth and are therefore candidates for a 16×16 prediction. $M =$ {input mode, all 16 × 16 modes};

b) G^H group (the highest 30% input bits consumption). Blocks are assumed to be highly detailed and their modification is expected to increase the coding efficiency. Therefore, we examine all 4 × 4 modes for this group. $M =$ {all 4 × 4 modes};

c) G^M group. $M =$ {input mode, 4 × 4 dc mode}.

2) HVS Characteristics Considerations: Psychovisual studies have led to the concept of a perceptual three-component image model [29]: texture regions, smooth regions, and edges. In [30], the authors suggest to modify the block's distortion value according to its perceptual importance, using six different perceptual groups, where each has a different f factor. The distortion is weighted by the $1/f$ factors and is plugged into the rate-distortion cost function. We follow this idea but segment the image into the three perceptual groups of texture regions, smooth regions, and edges. First, we calculate the variance of the block coefficients, where the dc term and the first two ac coefficients are not taken into account, to avoid slow intensity changes detection. The variances map is translated into low and high activity blocks using an adaptive threshold. Morphological operations are then used to detect the edges and smooth regions and form the segmented picture. Since artifacts are most apparent at smooth regions and less noticeable at textured regions, we set $f^{\text{texture}} > 1$, $f^{\text{smooth}} < 1$, and $f^{\text{edge}} = 1$. The specific parameter values are given in Section V-B.2.

IV. Inter-Frames Transrating

In Section I-B, we defined the closed-loop residual correction architecture for inter-frames, which also reuses the input motion decisions. Since the typical bit budget for inter-frames is low (as compared to intra-frames), the rate control should be accurate in order to meet the target bit rate. Therefore, we propose an optimal nonuniform requantization (Section IV-A). To reduce the computational load, we suggest using new macroblock level models, adapted to H.264 requantization (Section IV-B).

A. Optimal Requantization

1) Introduction: In previous standards, such as MPEG-2, the optimal requantization problem is defined as finding a set of optimal new step sizes that minimize the total distortion, subject to a given bit-rate constraint

$$\min_{\{QP_i\}} D, \quad \text{subject to: } R \leq R_{\text{target}} \quad (15)$$

where $D = \sum_{i=1}^{N_B} d_i(QP_i)$, $R = \sum_{i=1}^{N_B} r_i(QP_i)$, N_B is number of macroblocks in the frame, QP_i is quantization parameter for the ith macroblock, d_i is distortion caused to the ith macroblock, and r_i is number of bits produced by the ith requantized macroblock.

A common approach [1] is to convert the constrained optimization problem to an unconstrained one

$$\min_{\{QP_i\}} J, \quad J = D + \lambda(R - R_{\text{target}}) \quad (16)$$

where λ is the Lagrangian parameter. The main advantage of solving the unconstrained problem is that the cost J can be broken into a sum of independent costs for each macroblock. Given a λ value, the set of quantization steps $\{QP_i^*\}_{i=1}^{N_B}$ that minimizes the set of independent costs is found, and the corresponding average rate is calculated by $\sum_{i=1}^{N_B} r_i(QP_i^*)$. Then, the λ parameter is altered (e.g., using bisection iterations) until an average rate that is close enough to the target is obtained.

In [24], [30], [31], it is argued that avoiding large fluctuations in the quantization step size throughout the frame results in better subjective quality, as the overall perceived frame's quality appears constant and blocking artifacts are reduced. In

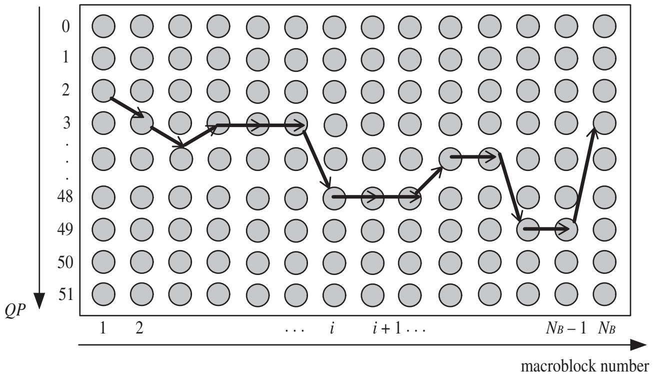

Fig. 7. Dynamic programming path illustration. Horizontal axis: macroblock number, vertical axis: quantization parameter. Each circle denotes a state and each column corresponds to a macroblock stage. The arrows show a path, where the change in QP from one macroblock to the next is within ±3 units.

addition, the H.264 standard encodes the quantization parameter differentially, that is, it encodes $\Delta QP = QP - QP_{\text{Prev}}$, where QP, QP_{Prev} are the quantization parameters of the current and the previous encoded macroblock according to a raster scan order. Moreover, the cost in bits of the ΔQP transition increases with its absolute value. As a result, many rate control algorithms for H.264 limit $|\Delta QP|$ to take small values (typically, up to 2).

2) Optimization: We assume that the change in the overhead bits due to transrating is negligible (see Section II) and define the optimization problem in terms of the texture bits

$$\min_{\{QP_i\}} J, \quad J = D + \lambda(R^{\text{texture}} - R^{\text{texture}}_{\text{target}}). \tag{17}$$

In addition, we propose to regulate the changes in QP to achieve better subjective quality by adding a regularization term $\mu \sum_{i=2}^{N_B} cost(\Delta QP_i)$, which accounts for the cost in bits of coding ΔQP (as defined in the standard [9]). As the weight parameter μ translates the regularization term measured in bits to distortion units and we do not try to achieve an exact bit target for coding ΔQP, we choose to set $\mu = \lambda$, so that it has the same units, simplifying the solution

$$\min_{\{QP_i\}} D + \lambda(R^{\text{texture}} - R^{\text{texture}}_{\text{target}}) + \lambda \sum_{i=2}^{N_B} cost(\Delta QP_i). \tag{18}$$

Since the choices of quantization step sizes for different macroblocks are no longer independent, the whole set of quantization step sizes $\{QP_i^*\}$ should be found at once. Therefore, we propose to extend each Lagrangian iteration with a dynamic programming stage. The external Lagrangian iterations change the Lagrangian parameter λ to improve the rate guess. At each examined value of λ, the dynamic programming algorithm finds an optimal QP path by solving (18), as will be explained next. The results showed that the above algorithm rarely chooses $|\Delta QP|$ values bigger than three. As there is no practical need for larger $|\Delta QP|$, we limit the allowed transition to $|\Delta QP| \leq 3$.

The optimization problem is then defined by

$$\min_{\{QP_i\}} D, \quad \text{subject to:} \tag{19}$$
$$R^{\text{texture}} \leq R^{\text{texture}}_{\text{target}} \text{ and } |\Delta QP| \leq 3.$$

At each examined value of λ, the constrained dynamic programming algorithm finds an optimal QP path by solving

$$\min_{\{QP_i\}} J, \text{ subject to: } |\Delta QP| \leq 3 \tag{20}$$

where $J = D + \lambda(R^{\text{texture}} - R^{\text{texture}}_{\text{target}}) + \lambda \sum_{i=2}^{N_B} cost(\Delta QP_i)$. The dynamic programming algorithm is defined over the set of states $\{(QP, i)\}$, where i is the macroblock index and QP is the quantization index, see Fig. 7. Each state (QP, i) has its cost value $j_i(QP) = d_i(QP) + \lambda r_i(QP)$ and the total frame's cost along a path is $J = \sum_{i=1}^{N_B} j_i(QP) + \lambda \sum_{i=2}^{N_B} cost(\Delta QP_i)$. The optimal path up to state (QP, i) is the path that has the minimal accumulated cost $V_i(QP^*)$ over all possible paths that end at that state. As $|\Delta QP| \leq 3$, there are at most seven possible paths that end at the previous macroblock $(i - 1)$ and can be continued to the current state (QP, i). We choose among these by minimizing the current state value function

$$V_i(QP) = V_{i-1}(QP_{\text{Prev}}) + j_i(QP) + \lambda cost(QP_{\text{Prev}}, QP) \tag{21}$$

where $QP_{\text{Prev}} - QP \in \{-3, -2, -1, 0, 1, 2, 3\}$. It is the sum of the cost of the path until the previous macroblock $V_{i-1}(QP_{\text{Prev}})$, the cost of the current state $j_i(QP)$, and the cost of moving from state $(QP_{\text{Prev}}, i - 1)$ to (QP, i). Or, in other words, the best path up to state (QP, i) is continued from state $(QP^*_{\text{Prev}}, i - 1)$, where

$$QP^*_{\text{Prev}} = \arg\min_{QP_{\text{Prev}}} \{V_{i-1}(QP_{\text{Prev}}) + \lambda cost(QP_{\text{Prev}}, QP)\}. \tag{22}$$

The corresponding value function update is then $V_i(QP) = V_{i-1}(QP^*_{\text{Prev}}) + j_i(QP) + \lambda cost(QP^*_{\text{Prev}}, QP)$.

At each stage i of the dynamic programming algorithm (from the first to the last macroblock), the best paths for all (QP, i) states are found and kept as lists of pointers, along with their values. At the last stage $(i = N_B)$, the best path found is the optimal path over the entire frame

$$PathEnd^* = \arg\min_{QP} V_{N_B}(QP). \tag{23}$$

The algorithm then traces back the best frame path using the chosen list of pointers, to obtain the optimal path: $\{QP_i^*\}_{i=1}^{N_B}$. Since we would like to reduce the bit rate, we constrain the requantized step sizes not to be finer than the original step-sizes. Thus, states that correspond to QP smaller than the original are assigned an infinite cost and discarded from the search procedure. The dynamic programming algorithm is performed at each Lagrangian iteration. The Lagrangian iteration convergence criterion is that the resulting rate deviates from the target rate by no more than 4%. In addition, in case the bisection algorithm is stuck, there is also a tolerance of 0.1% on the minimal amount of change in λ between consecutive Lagrangian iterations. The number of Lagrangian iterations required until convergence is 6 to 8, on average.

3) Coefficient Elimination: After applying the transform and quantization, the quantized index blocks are typically sparse. At the encoder, or the transcoder for that matter, it is possible to modify the obtained index levels to achieve a lower cost, in terms of rate-distortion. In [32]–[34], index

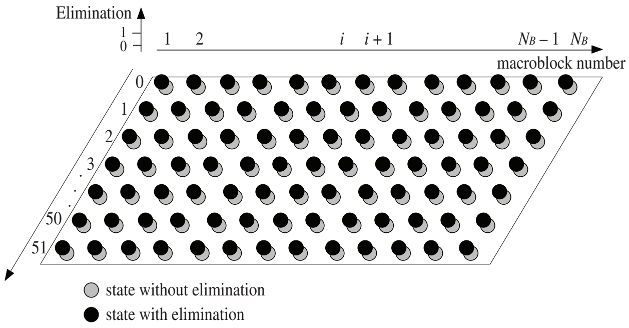

Fig. 8. Two-layer array illustration. Horizontal axis: macroblock number, vertical axis: quantization parameter. Each disc denotes a state. Black and gray colors correspond to states with and without elimination, respectively.

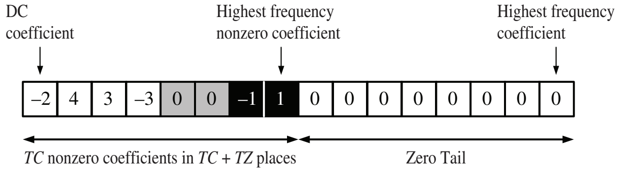

Fig. 9. Example of the additional overhead syntax elements in H.264 for a zig-zag scanned block. There are $TC = 6$ nonzero coefficients, and $TZ = 2$ zeros (marked in gray) counted from the dc coefficient to the highest frequency nonzero coefficient. The trailing ones are marked in black.

modification was examined by evaluating the modified costs exhaustively, that is, evaluating a few optional rates directly from the entropy-coding tables *without using models*. A simpler case of index modification is coefficient elimination, or thresholding [35]–[37]. Specifically, [37] considers the coefficient elimination rule used in the H.264-recommended encoder. It zeroes sparse blocks that are almost zeroed except for a few trailing ones (±1 at the end of the block) corresponding to transform coefficients at high frequencies.

We examined incorporating selective coefficient elimination into the proposed rate-distortion optimization algorithm. To reduce the computational load regarding which coefficient to eliminate, we follow the simple elimination rule used in the recommended encoder software. We optimally decide for each quantized MB whether to encode it as is or to perform coefficient elimination first, as follows. Two rate-distortion pairs are evaluated for each combination of quantization parameter QP and macroblock index i: $\{d_i^0(QP), r_i^0(QP)\}$ and $\{d_i^1(QP), r_i^1(QP)\}$, for the cases of no elimination and elimination, respectively. As a result, a two-layer array for the rate and the distortion is generated over the set of states $\{(QP, i, elim)\}$, where elim $\in \{0, 1\}$ is a binary flag that denotes whether or not elimination is performed (see Fig. 8). The optimization problem is then defined by

$$\min_{\{QP_i, elim_i\}} D, \text{ subject to:} \qquad R \leq R_{\text{target}} \quad \text{and} \quad |\Delta QP| \leq 3 \tag{24}$$

where $D = \sum_{i=1}^{N_B} d_i^{\text{elim}_i}(QP_i)$ and $R = \sum_{i=1}^{N_B} r_i^{\text{elim}_i}(QP_i)$.

To solve the optimization problem of (24), we follow the Lagrangian iterations extended by a dynamic programming algorithm, where the latter is then extended from a single 2-D layer to two layers. At the last stage ($i = N_B$), the best path $\{QP_i^*, elim_i^*\}_{i=1}^{N_B}$ is the optimal path over the entire frame

$$(PathEnd^*, ElimEnd^*) = \arg \min_{QP, \text{elim}} V_{N_B}^{\text{elim}_{N_B}}(QP). \tag{25}$$

We compared the performance of the selective coefficient elimination with that of no elimination, where in both cases the requantization step sizes were optimally selected. The current selective elimination implementation shows a small gain in terms of PSNR versus bit rate (about 0.07 dB), as only a small part of the frame blocks is selected for elimination. Full elimination (without selection) is not recommended and the PSNR loss at high bit rates can get to 0.4 dB. Still, we believe that this algorithm can potentially achieve a higher gain, by using more sophisticated elimination rules.

B. Rate-Distortion Modeling

The optimization algorithm described above requires evaluating the rate and distortion obtained by requantizing each macroblock at multiple step sizes. If no prior knowledge is used, such rate assessment involves performing actual requantization followed by entropy coding multiple times. The optimization then becomes computationally expensive. The computational complexity can be greatly reduced by using an analytic model for the relation between rate and quantization step size, *for each macroblock*. We suggest [10] modified models for H.264 at the macroblock level. Since the proposed rate–ρ model is especially adapted for requantization in the H.264 standard, we briefly outline the H.264 entropy coding first and then describe the proposed model.

1) H.264 Context Adaptive Entropy Coding: The H.264 context adaptive entropy coding with variable length coding (CAVLC) tables, is designed to take advantage of the sparse (compact energy) characteristics of the quantized transform coefficients [4]. To this end, it uses a set of syntax elements that includes both the customary run-level representation and additional overhead counts that mainly describe the zero valued coefficients distribution. On top of that, it switches between several VLC tables for each syntax element, in a context adaptive manner.

Though the run and level are encoded separately, their encoding is efficient due to the context-based VLC table switching. The additional overhead counts consist of two symbols. One describes the combination of the number of nonzero coefficients and the high-frequency trailing ones (±1 at the end of the block). We shall refer to it as (TotalCoefficients, TrailingOnes). The other symbol, called TotalZeros, denotes the number of zeroed coefficients from the dc coefficient to the highest frequency nonzero coefficient, both of which use multiple VLC tables. Fig. 9 shows an example for a 4 × 4 zig-zag scanned block, with six nonzero coefficients, two trailing ones (marked in black), and two TotalZeros (marked in gray).

2) Rate–ρ Model for H.264 Requantization: Examination of the rate–ρ relation at the macroblock level has shown that a linear relation is not a good descriptor of the empirical data. Therefore, and in light of H.264 new entropy coding features, we suggest a new rate–ρ model at the macroblock level. We decompose the rate into *data* and *overhead* components,

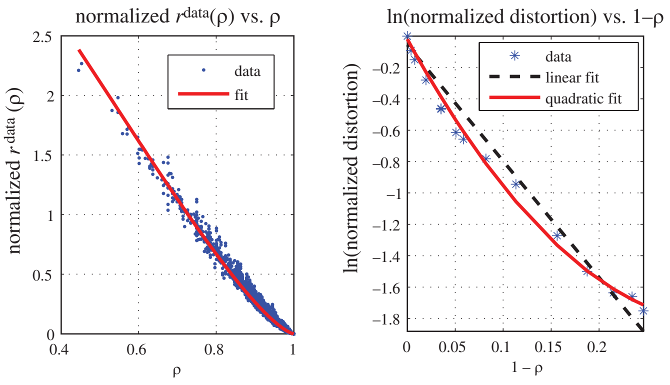

Fig. 10. ρ-domain models fits. Left: $r^{\text{data}}(\rho)$, blue dots: normalized $r^{\text{data}}(\rho)$ relation of one frame's macroblocks; red solid line: the fit with the common shape parameter η. Here, $\eta = 1.38$ and $\bar{\theta} = 6.5$. Right: $d(\rho)$ model fit example, blue points: $ln(\overline{d}(\rho))$; black dashed line: linear fit; red solid line: quadratic fit. Data is from the *Flower Garden* sequence.

where the data stands for the bits spent on coding the run-level, and the overhead designates the bits spent on coding the new syntax elements. For the model parameter estimation we use prior information, such as the original input quantized transform coefficients and their encoded rate.

a) Data component: The data texture bits component is composed of coding the run-level syntax elements that form the majority of the texture bits at moderate to high bitrates. This component rate–ρ relation is a monotonically decreasing convex function.

Therefore, for the data component rate–ρ relation, we suggest the following closed-form model

$$r^{\text{data}}(\rho) = \theta \cdot ln(1 + (1-\rho)^{\eta}) \tag{26}$$

where $\theta \geq 0$, $\eta \geq 1$. The parameter θ controls the scale of the curve, whereas the parameter η changes its shape. Now, given this component's original input encoded rate of a macroblock, $r_{\text{in}}^{\text{data}}(\rho_{\text{in}})$, we can fit one of the parameters. Since this model requires fitting two parameters, we apply a 2-D search to fit its shape parameter η and an average scale parameter $\bar{\theta}$ using the input ensemble $\{r_{\text{in},i}^{\text{data}}(\rho_{\text{in},i})\}_{i=1}^{N_B}$ of all the frame macroblocks. The estimated shape parameter η is used for all the frame macroblocks. The scale parameter θ_i is then matched to each macroblock separately by (27). Luminance and chrominance components are modeled separately

$$\theta_i = \frac{r_{\text{in},i}^{\text{data}}}{ln(1 + (1-\rho_{\text{in},i})^{\eta})}. \tag{27}$$

Since the frame macroblocks share the same parameter η, but each has a different parameter θ_i, we cannot depict their model-based fittings on a single graph. However, we can scale all macroblock level relations using the average frame level parameter $\bar{\theta}$, by drawing $r_i^{\text{data}}(\rho_i) \cdot (\bar{\theta}/\theta_i)$ and then drawing their common fit $r^{\text{data}}(\rho_i) = \bar{\theta} \cdot ln(1 + (1-\rho_i)^{\eta})$. On the left of Fig. 10 the scaled rate-ρ relation of each macroblock is denoted by blue dots and the common fit by a red line.

b) Overhead component: The overhead component is composed of coding the (TotalCoefficients, TrailingOnes), and TotalZeros syntax elements, denoted for short by (TC, TR) and TZ, respectively. This rate–ρ relation is very noisy due to two reasons. One is that the syntax elements values (e.g., $(TC, TR) = (6, 2)$ and $TZ = 2$ in the example of Fig. 9) are not uniquely defined by the local block's ρ. The other is the use of multiple VLC tables for each syntax element, which means that the number of bits spent on coding the same syntax element value changes with the context. As a result, fitting a closed-form model for it becomes practically impossible. However, there is a partial dependency on ρ, as the macroblock's level ρ is the average of ρ_b's: $\rho = (1/16)\sum_{b=1}^{16} \rho_b$, and $\rho_b = 1 - [(TC_b)/16]$, where ρ_b is the local ρ for a 4 × 4 block and TC_b is the blocks's TC count. Using the statistical model that follows, we calculate once the average code lengths $\bar{c}_{(TC,Tr)}(\rho_b|context\ prior)$ and $\bar{c}_{TZ}(\rho_b|input\ prior)$ of the (TC, TR) and TZ syntax elements, respectively. These average lengths are kept in lookup tables and the rate "overhead" component is obtained by averaging over all the blocks in the macroblock

$$\begin{aligned} r^{\text{overhead}}(\rho) = {} & \frac{1}{16}\sum_{b=1}^{16} \bar{c}_{(TC,Tr)}(\rho_b|context\ prior) \\ & + \frac{1}{16}\sum_{b=1}^{16} \bar{c}_{TZ}(\rho_b|input\ prior). \end{aligned} \tag{28}$$

We assume that the quantized transform coefficients are not correlated and follow a Laplacian distribution. Another assumption is that all ±1 quantized coefficient appearances occur at the highest nonzero frequencies, and are thus considered as high-frequency trailing ones. Using the Laplacian distribution, the probability that the magnitude of a quantized transform coefficient, l, will take the value k is

$$Pr.(|l| = k) = \begin{cases} \rho & k = 0 \\ \frac{(1-\rho)^{2k}\rho(2-\rho)}{1-\rho} & k > 0. \end{cases} \tag{29}$$

Hence the probability of a trailing-one coefficient, given that it is non-zero is $Pr.(TR) = Pr.(|l| = 1 \,|\, |l| > 0) = \rho(2-\rho)$.

We define a binomial random variable that denotes the number of trailing-one appearances given ρ_b and sum over the joint (TC, TR) code length tables (four different tables) to obtain the average VLC tables $\bar{c}_{(TC,Tr)}(\rho_b|context\ prior)$. We switch between these four average VLC tables by predicting the number of nonzero coefficients from the neighboring blocks, in accordance with the standard's context-based encoding.

Since the quantized blocks are typically sparse and most of the energy is concentrated at low frequencies, there is usually a tail of zeros at the end of the scanned block (see example in Fig. 9). So, instead of counting TZ, the number of zeroed coefficients from the dc coefficient to the highest frequency nonzero coefficient we can count its complement, the tail, since $TC + TZ + Ztail = 16$. As we increase the requantization step, the number of nonzero coefficients TC decreases, and the tail length monotonically increases. Therefore, $TC + TZ$ monotonically decreases. Given the input prior information $(TC_{\text{in}}, TZ_{\text{in}})$, we find the probability of having TZ TotalZeros given ρ_b. The average code length for each of the 15 $(TC_{\text{in}}, TZ_{\text{in}})$ input priors is evaluated by summing over the joint (TC, TZ) code length tables. Finally, the total rate–ρ relation is evaluated by (30) where $r^{\text{data}}(\rho)$ and $r^{\text{overhead}}(\rho)$ are evaluated from (26) and (28), respectively

$$r(\rho) = r^{\text{data}}(\rho) + r^{\text{overhead}}(\rho). \tag{30}$$

3) Distortion$-\rho$ Model: The PSNR is a widely used objective quality metric that is related to the MSE distortion. That is why we examined the validity of the exponential *distortion–ρ* model suggested in [3] in describing the MSE. According to this model, $ln(\overline{d}(\rho))$ should be linearly proportional to $1-\rho$, where $\overline{d}(\rho) = [d(\rho)]/\sigma^2$ is the normalized distortion. Examining this relation at the macroblock level, we found that a linear model does not describe it with sufficient accuracy. We therefore suggest extending the model to an exponential-quadratic relation (31) that better matches the empirical data (see the right of Fig. 10) and a quantitative accuracy comparison on the bottom of Fig. 12

$$d(\rho) = \sigma^2 \cdot e^{\alpha_1\cdot(1-\rho)^2+\alpha_2\cdot(1-\rho)}. \tag{31}$$

The modified *disortion–ρ* model has three parameters that need to be estimated: α_1, α_2, and σ^2. Since we do not have the signal at the input of the first encoder, we can only measure the requantization distortion, and not the total degradation from the reference. The scale parameter σ^2 is calculated once as the sum of squares of the input transform coefficients, as this would be the MSE if the block is zeroed. Given σ^2, we evaluate the normalized distortion $\overline{d}(\rho)$ for two different requantization step sizes and extract the α_1, α_2 parameters [38]. The luminance and chrominance components are modeled separately.

4) ρ–Q_2 Relation: Contrary to intra-coded frames, the estimation of ρ for inter-coded frames is fairly simple and has low complexity. Since the inter-coded blocks are predicted using previously decoded frames, their closed loop correction signal is available and the model evaluation is performed based on the corrected transform coefficients to be requantized. So, we count the number of coefficients that fall in the second quantizer deadzone $[-t(Q_2), t(Q_2)]$, where $t(Q_2) = (1-\Delta z)Q_2$ and Δz is the deadzone. The ρ–Q_2 relation is evaluated using this histogram count by normalizing the expected number of zeros at the quantizer output to the data size. It is evaluated for each macroblock for all step sizes coarser than the input step size, prior to the rate and the distortion evaluation. When the selective elimination algorithm is applied, ρ is evaluated by applying the same histogram count on the quantized coefficients after elimination.

V. Results

In this section, we summarize and report the main simulation results of the developed algorithm. The original video sequences were first encoded at 2 Mb/s using H.264 baseline profile and then transrated at four transrating ratios. The standard video sequences used for the analysis are *Flower Garden*, *Football*, *Mobile and Calendar* at SIF format (352 × 240 resolution) and *Foreman* at a CIF format (352 × 288 resolution). We also examined the *Pedestrian* sequence at SDTV format (720 × 576 resolution) originally encoded at 8 Mb/s.

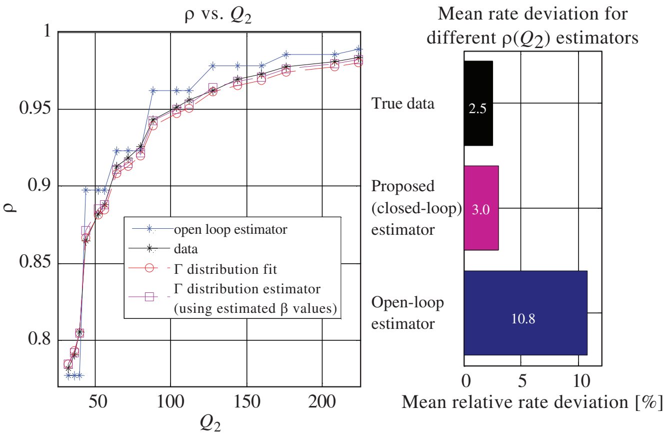

Fig. 11. Intra-models accuracy. Left: frame level $\rho(Q_2)$ relation (from the *Flower Garden* sequence), blue staircase line: open-loop estimator; black asterisk: Data; red circles: proposed estimator (offline β evaluation using ML); magenta squares: proposed estimator (using estimated β values). Right: Mean relative rate deviation from the target, measured for the four examined sequences initially encoded at 2 Mb/s, at intra transrating factors of 1.5 to 3.

A. System Architecture

The chosen architecture is FD-GE for intra-frames and PD-PE for inter-frames (Section I-B). The PD-PE architecture reduces the runtime of inter-frame transrating by about 15% as compared to a FD-GE architecture, at negligible quality loss. If the FD-GE architecture is used for inter-frames too, the input motion vectors (MVs) could be modified. Our attempt to modify the input MVs by locally merging them has shown that a further MV refinement search is required to avoid quality degradation. Since such a refinement further increases the computational complexity, we chose to reuse the input motion decisions. Another extension of our work using the FD-GE architecture for inter frames is discussed in Section V-E.

B. Intra-Frames Transrating

1) Model-Based Uniform Requantization: In Section III-A.2, we proposed a closed-loop statistical model for estimating the $\rho(Q_2)$ relation for an intra-frame. An example for this $\rho(Q_2)$ estimator at the frame level, as compared to other estimators, is depicted on the left of Fig. 11. The open-loop estimator is biased as compared to the true data relation and, as noted earlier, has a staircase characteristic. The proposed estimators follow closely the data, and their average relative error is less than 1.7%. We examined the average rate deviation from the target, where the uniform requantization step size was selected using different $\rho(Q_2)$ estimators, as depicted on the right of Fig. 11. The true data $\rho(Q_2)$ relation was used as a yardstick for the performance, as it cannot be evaluated in a real-time scenario. It shows a small rate estimation error (2.5%), mainly due to the rate–ρ model's inaccuracy. Since the open-loop estimator is inherently biased, it tends to choose finer steps than required, resulting in a higher rate and a large rate estimation error. The proposed $\rho(Q_2)$ estimator outperforms the open-loop estimator, providing a smaller error, close to the rate estimation error from the true data.

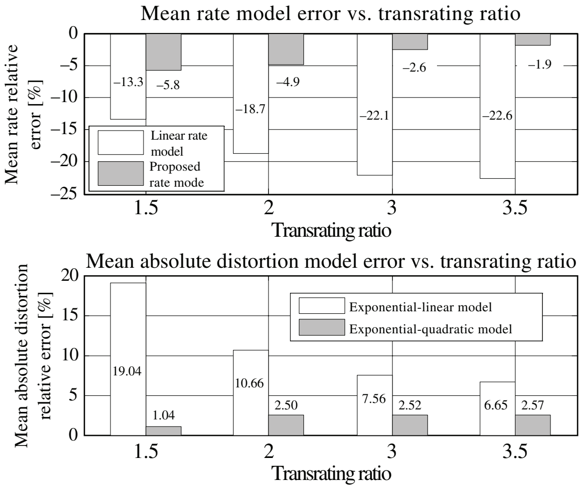

Fig. 12. Rate and distortion models accuracy. Top: mean rate-model error versus transrating ratio, white: linear rate model, gray: proposed rate model. Bottom: mean absolute distortion error versus transrating ratio, white: exponential-linear model, gray: exponential-quadratic model.

2) Modification of Prediction Modes: In Section III-B.2, we considered weighting the distortion of texture regions, smooth regions, and edges differently. Since HVS-based considerations are not in the main focus of our work, the weighting factors were set empirically to $f^{\text{texture}} = 1.2$, $f^{\text{smooth}} = 0.8$, and $f^{\text{edge}} = 1$. The visual effect of the prediction modes modification is more noticeable at smooth regions, e.g., the sky in the *Flower Garden* sequence. Reusing the input prediction modes reduces the runtime of intra-frame transrating by a factor of about 4.5, on average, as compared to re-encoding, at a PSNR loss of up to 1 dB (for a transrating factor of three). The proposed selective mode modification scheme, suggested in Section III-B, has practically the same performance as the intra-frame re-encoding scheme in terms of PSNR versus bit rate, at about 37.5% less computations. By comparison, reuse of input modes is faster and more suitable for small transrating factors, as the transrated frame prediction modes are expected to be similar to the input modes.

C. Inter-Frames Transrating

The motivation for using the rate-distortion models proposed in Section IV-B is to provide an accurate and low computational rate-distortion evaluation. We now discuss the performance of the proposed MB-level models in terms of accuracy and computational complexity. The mean rate-model error, for both the proposed and the linear rate models, measured as the deviation of the model-based rate estimation from the actual encoded number of bits is depicted in the upper part of Fig. 12. The proposed rate–ρ model errors are smaller than the linear rate–ρ model errors. As the bit rate is reduced, the "overhead" component in the rate model gets more dominant and more accurate. As a result, the overall accuracy of the proposed model is improved for higher transrating ratios. The accuracy of the exponential-linear distortion–ρ model suggested in the literature is compared with the proposed exponential-quadratic model on the bottom of Fig. 12. It shows an average error of only 2% for the proposed exponential-quadratic model versus 11% for the exponential-linear model. To evaluate the computational complexity of the inter-frames transrating, its two phases should be considered: model-based rate-distortion evaluation, and the optimization procedure. We compared the runtime of inter frame transrating when an optimal requantization is performed (see Section IV-A), once using the proposed rate-distortion models and once using an exhaustive rate-distortion evaluation (i.e., without models). By evaluating the proposed rate-distortion models, the runtime is reduced by a factor of about 2.3, on average, as compared to the exhaustive evaluation. As for the optimization procedure complexity, it takes about six to eight Lagrangian iterations until convergence. Each such Lagrangian iteration requires $MB_{\text{num}} QP_{\text{num}}$ basic operations of finding the best previous value (minimum of a 7-length array), where MB_{num} is the number of macroblocks in the frame and $QP_{\text{num}} = 52$.

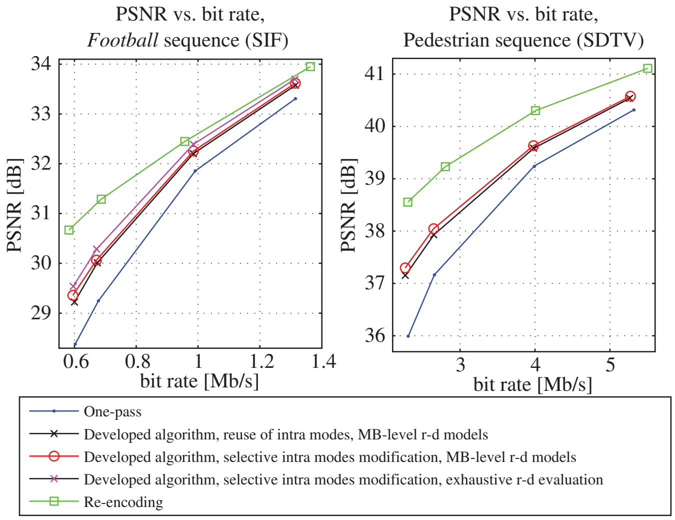

Fig. 13. PSNR versus bit rate. Left: transrating a SIF *Football* sequence, initially encoded at 2 Mb/s. Right: transrating a SD *Pedestrian* sequence, initially encoded at 8 Mb/s. Blue dots: one-pass requantization. Black x: developed algorithm, reuse of intra modes, MB-level R-D models. Red circles: developed algorithm, selective intra modes modification, MB-level R-D models. Magenta x: developed algorithm, selective intra modes modification, exhaustive R-D evaluation. Green squares: re-encoding.

D. Overall System Performance

We summarize and compare by simulations the following transrating algorithms:

1) Re-encoding;
2) Proposed algorithm;
3) One-pass requantization [38] (sets the requantization step size of 1 MB at a time, according to the output buffer fullness. For fair comparison, it also uses the optimal GOP level bit allocation suggested in Section II).

SIF, *Football* SIF, *Mobile and Calendar* SIF and *Foreman* CIF) were first encoded at 2 Mb/s using H.264 baseline profile, with a GOP structure of 1 I-frame followed by 14 P-frames and no frame skipping allowed. The encoding was done using Nokia H.264 baseline encoder. These were then transrated at four transrating ratios. The PSNR versus bit rate graph for the *Football* sequence is depicted in on the

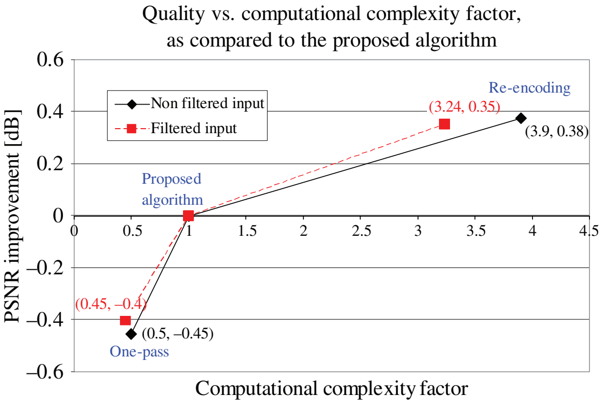

Fig. 14. Quality versus computational complexity of re-encoding and one-pass algorithm, as compared to the proposed algorithm, for SIF/CIF resolution sequences. The quality is measured by PSNR improvement, and the computational complexity is measured by the runtime factor. Black solid diamond: Input encoded without deblocking filter, Red dashed square: Input encoded with deblocking filter.

left of Fig. 13. As expected, it rates the performance of the transrating algorithms at the following order, from the best to the worst quality: re-encoding, the proposed algorithm, and one-pass requantization. It also shows that the selective intra-modes modification (denoted by red circles) has better performance than reusing the input intra-modes (denoted by black x). It should be noted that for fair comparison, the model-based optimal GOP level bit allocation was applied for the one-pass method too, but it is more likely that such a simple requantization would use a simpler GOP allocation as well, which would further decrease its performance. The same ranking of the algorithms in terms of PSNR versus bit rate turned out at other cases, e.g., transrating *Football* SIF initially encoded at 1 Mb/s and *Foreman* QCIF initially encoded at 250 kb/s. We also compared the tested algorithm at SDTV resolution by transrating the SD *Pedestrian* sequence originally encoded at 8 Mb/s, as depicted on the right of Fig. 13. The results show the same algorithms ranking concluded from our previous experiments at lower spatial resolutions, with larger PSNR gaps.

The overall system performance is measured in terms of computational complexity (by runtime) and quality (by the PSNR difference). The quality versus computational complexity, for the different algorithms, as compared to the proposed algorithm, is depicted by the black solid curve in Fig. 14. The graph shows average results over four video sequences encoded at 2 Mb/s and transrated to 1 Mb/s. As compared to re-encoding, the proposed algorithm saves the runtime by a factor of about four, on average, with small PSNR loss at high to medium bit rates. In comparison with a simple one-pass requantization, the proposed algorithm achieves better performance, at the cost of twice the complexity. In [6], the authors compare their algorithm with re-encoding and report on saving a factor of about two in the runtime at a PSNR loss of about 0.5 dB, which is worse than our proposed system performance. By examining the graph slopes in Fig. 14, we conclude that the proposed system's gain, as compared to the one-pass requantization, is higher than the re-encoding gain as compared to the proposed system.

E. Support of Input Coded With Deblocking Filter

H.264 may apply an adaptive in-loop deblocking filter on the decoded pictures to reduce blocking artifacts [39]. However, it is not clear whether the computational cost of the filter is justified considering the improvement in subjective quality [40]. In this paper, we assumed that the deblocking filter was disabled during encoding of the input video and its transrating. To support input video that was initially encoded using the deblocking filter, we propose to fully decode the input (including the in-loop filtering) and then encode according to our algorithm without applying the filter. To evaluate the performance, we ran again the tests described in Section V-D for an input video initially encoded with the deblocking filter (see red dashed curve in Fig. 14). Here, the proposed system runtime increases due to the decoding with a deblocking filter and therefore the complexity saving factor as compared to re-encoding is somewhat reduced. Still, the proposed system provides a good tradeoff between quality and computational complexity.

VI. Conclusion

A model-based transrating system for H.264 encoded video via requantization has been proposed. To keep a smooth constant video quality, it applies an optimal GOP level bit allocation that equalizes the frame distortions. For intra-coded frames, a uniform requantization step size is chosen using the linear rate–ρ model and a novel closed-loop statistical estimator for the ρ–Q_2 relation. This estimator overcomes the spatial-block dependency problem by modeling the correction signal of the requantized residual. For the examined sequences, its average rate deviation from the target is 3%, as compared to 10.8% average deviation obtained by using an open-loop ρ–Q_2 estimator. The guided intra-frame transrating allows either reuse of the input prediction modes, or their selective modification, reducing the computational complexity. For inter-coded frames, a new optimal nonuniform requantization algorithm is developed, where the changes in the requantization step sizes throughout the frame are regulated, to improve the subjective quality. To reduce that optimization computational burden, we suggest new macroblock level rate-distortion models in the ρ-domain, adapted to H.264 requantization. The incorporation of these models reduces the runtime of inter-frames transrating by a factor of about four, on average, with only a small PSNR loss at high to medium bit rates, for SIF/CIF resolution sequences.

References

[1] P. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Trans. Circ. Syst. Video Tech.*, vol. 8, no. 8, pp. 953–967, Dec. 1998.

[2] J. Lan, W. Zeng, and X. Zhuang, "Operational distortion-quantization curve-based bit allocation for smooth video quality," *Signal Process.: Image Commun.*, vol. 16, no. 4, pp. 527–543, Aug. 2005.

[3] Z. He and S. Mitra, "Optimum bit allocation and accurate rate control for video coding via ρ-domain source modeling," *IEEE Trans. Circ. Syst. Video Tech.*, vol. 12, no. 10, pp. 840–894, Oct. 2002.

[4] I. Richardson, *H.264 and MPEG-4 Video Compression*. New York: Wiley, 2003.

[5] P. Zhang, Q. Huang, and W. Gao, "Key techniques of bit rate reduction for H.264 streams," in *Proc. PCM 2004*, LNCS vol. 3332. pp. 985–992.

[6] H. Nam, “Low complexity H.264 transcoder for bitrate reduction,” in *Proc. ISCIT*, Bangkok, Thailand, Oct. 2006, pp. 679–682.

[7] D. Lefol, D. Bull, and N. Canagarajah, “An efficient complexity-scalable video transcoder with mode refinement,” *Signal Process.: Image Commun.*, vol. 22, no. 4, pp. 421–433, Apr. 2007.

[8] Z. He and S. Mitra, “A linear source model and a unified rate control algorithm for DCT video coding,” *IEEE Trans. Circ. Syst. Video Tech.*, vol. 12, no. 11, pp. 970–982, Nov. 2002.

[9] *H.264 Reference Software* [Online]. Available: http://bs.hhi.de/simsuehring/tml/download/

[10] N. Hait and D. Malah, “Toward model-based transrating of H.264 coded video,” in *Proc. IEEE 24th Conv. Elect. Electron. Engineers*, Eilat, Israel, Nov. 2006, pp. 133–137.

[11] N. Hait and D. Malah, “Model-based transrating of H.264 intra-coded frames,” in *Proc. PCS*, Lisbon, Portugal, Nov. 2007.

[12] T. Wiegand, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circ. Syst. Video Tech.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[13] Z. He, Y. Kim, and S. Mitra, “Low-delay rate control for DCT video coding via ρ-domain source modeling,” *IEEE Trans. Circ. Syst. Video Tech.*, vol. 11, no. 8, pp. 928–940, Aug. 2001.

[14] S. Milani, L. Celetto, and G. Mian, “A rate control algorithm for the H.264 encoder,” in *Proc. Workshop Signal Process. Commun.*, Baiona, Spain, Sep. 2003, pp. 295–300.

[15] I. Shin, Y. Lee, and H. Park, “Rate control using linear rate-ρ model for H.264,” *Signal Process.: Image Commun.*, vol. 19, no. 4, pp. 341–352, Apr. 2004.

[16] H. Sun, X. Chen, and T. Chiang, *Digital Video Transcoding for Transmission and Storage*. Boca Raton, FL: CRC Press, 2005.

[17] I. Ahmad, “Video transcoding: An overview of various techniques and research issues,” *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 793–804, Oct. 2005.

[18] Z. Lei and N. Georganas, “Rate adaptation transcoding for precoded video streams,” in *Proc. ACM Int. Conf. Multimedia*, Juan-les-Pins, France, Dec. 2002, pp. 127–136.

[19] 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.

[20] A. Vetro, J. Cai, and C. Chen, “Rate-reduction transcoding design for wireless video streaming,” *Wireless Commun. Mobile Computing*, vol. 2, no. 6, pp. 625–641, Oct. 2002.

[21] H. Sun, W. Kwok, and J. Zdepski, “Architectures for MPEG compressed bistream scaling,” *IEEE Trans. Circ. Syst. Video Tech.*, vol. 6, no. 2, pp. 191–199, Apr. 1996.

[22] D. Lefol, D. Bull, and N. Canagarajah, “Performance evaluation of transcoding algorithms for H.264,” *IEEE Trans. Consumer Electron.*, vol. 52, no. 1, pp. 215–222, Feb. 2006.

[23] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[24] M. Militzer, M. Suchomski, and K. Meyer-Wegener, “Improved ρ-domain rate control and perceived quality optimizations for MPEG-4 real-time video applications,” in *Proc. ACM Int. Conf. Multimedia*, 2003, pp. 402–411.

[25] Y. Altunbasak and N. Kamaci, “ρ domain rate-distortion optimal rate control for DCT-based video coders,” in *Proc. ICASSP*, Montreal, Canada, May 2004, pp. 149–152.

[26] C. Chen, P. Wu, and H. Chen, “Transform-domain intra prediction for H.264,” in *Proc. ISCAS*, May 2005, pp. 1497–1500.

[27] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. 2nd ed. New York: McGraw-Hill, 1986.

[28] D. Lefol, D. Bull, and N. Canagarajah, “Mode refinement algorithm for H.264 intra frame requantization,” in *Proc. ISCAS*, Island of Kos, Greece, 2006, pp. 4459–4462.

[29] L. Torres and M. Kunt, *Video Coding: The Second Generation Approach*. Norwell, MA: Kluwer, 1996, ch. 6.

[30] K. Minoo and T. Nguyen, “Perceptual video coding with H.264,” in *Proc. IEEE Conf. Signals, Syst. Comput.*, Pacific Grove, CA, 2005, pp. 741–745.

[31] A. Nguyen and J. Hwang, “A novel hybrid HVPC/mathematical model rate control for low bit-rate streaming video,” *Signal Process.: Image Commun.*, vol. 17, no. 5, pp. 423–440, May 2002.

[32] M. Lavrentiev, “Transrating of coded video signals via optimized requantization,” M.Sc. thesis, Dept. Elect. Eng., Technion Univ., Haifa, Israel, 2004.

[33] M. Lavrentiev and D. Malah, “Transrating of MPEG-2 coded video via requantization with optimal trellis-based DCT coefficients modification,” in *Proc. EUSIPCO*, Sep. 2004, pp. 1963–1966.

[34] W. Wang, H. Cui, and K. Tang, “Rate distortion optimized quantization for H.264/AVC based on dynamic programming,” in *Proc. SPIE Visual Comm. Image Process.*, vol. 5960, Jul. 2005, pp. 2100–2111.

[35] R. Lagendjik, E. Frimout, and J. Biemond, “Low-complexity rate-distortion optimal transcoding of MPEG I-frames,” *Signal Process.: Image Commun.*, vol. 15, no. 6, pp. 531–544, Mar. 2000.

[36] A. Eleftheriadis and D. Anastassioiu, “Constrained and general dynamic rate shaping of compressed digital video,” in *Proc. ICIP*, Washington, D.C., 1995, pp. 396–399.

[37] P. Carlsson, F. Pan, and L. T. Chia, “Coefficient thresholding and optimized selection of the lagrangian multiplier for non-reference frames in H.264 video coding,” in *Proc. ICIP*, 2004, pp. 773–776.

[38] N. Hait, “Model-based transrating of coded video,” M.Sc. thesis, Dept. Elect. Eng., Technion Univ., Haifa, Israel, 2007. [Online]. Available: http://sipl.technion.ac.il/siglib/FP/Hait.pdf

[39] P. List, “Adaptive deblocking filter,” *IEEE Trans. Circ. Syst. Video Tech.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.

[40] Y. Zhong, “Perceptual quality of H.264/AVC deblocking filter,” in *Proc. IEE Int. Conf. Visual Inform. Eng.*, Apr. 2005, pp. 379–384.

Naama Hait (S’06–M’08) received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 2003 and 2007, respectively.

She joined Elbit Systems Ltd. in 2008. Her research interests include video coding and transcoding, and image processing.

David Malah (S’67–M’71–SM’84–F’87–LF’09) received the B.Sc. and M.Sc. degrees in 1964 and 1967, respectively, from the Technion-Israel Institute of Technology, Haifa, Israel, and the Ph.D. degree in 1971 from the University of Minnesota, Minneapolis, all in electrical engineering.

Following one year on the staff of the Electrical Engineering Department of the University of New Brunswick, Fredericton, NB, Canada, he joined Technion in 1972, where he is an Elron-Elbit Professor of electrical engineering. During the period 1979 to 2001, he spent about six years, cumulatively, of sabbaticals and summer leaves at AT&T Bell Laboratories, Murray Hill, NJ, and AT&T Laboratories, Florham Park, NJ, conducting research in the areas of speech and image communication, and the summer of 2004 at Georgia Centers for Advanced Telecommunications Technology-GCATT, Georgia Institute of Technology, working in the area of video processing. Since 1975, he has been the academic head of the Signal and Image Processing Laboratory, Technion, which is active in image/video and speech/audio processing research and education. His main research interests are in image, video, speech, and audio coding, speech and image enhancement, hyperspectral image analysis, data embedding in signals, and digital signal processing techniques.