No-Reference Video Quality Assessment using Codec Analysis

Søgaard, Jacob; Forchhammer, Søren; Korhonen, Jari

Published in:
I E E E Transactions on Circuits and Systems for Video Technology

Link to article, DOI:
10.1109/TCSVT.2015.2397207

Publication date:
2015

Document Version
Peer reviewed version

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2015.2397207, IEEE Transactions on Circuits and Systems for Video Technology

No-Reference Video Quality Assessment using Codec Analysis

Jacob Søgaard, Søren Forchhammer, *Member IEEE*, and Jari Korhonen, *Member IEEE*
DTU Photonics
Technical University of Denmark
Kgs. Lyngby, Denmark

***Abstract*—A no-reference video quality assessment (VQA) method is presented for videos distorted by H.264/AVC and MPEG-2. The assessment is performed without access to the bit-stream. Instead we analyze and estimate coefficients based on decoded pixels. The approach involves distinguishing between the two types of videos, estimating the level of quantization used in the I-frames, and exploiting this information to assess the video quality. In order to do this for H.264/AVC, the distribution of the DCT-coefficients after intra-prediction and deblocking are modeled. To obtain VQA features for H.264/AVC, we propose a novel estimation method of the quantization in H.264/AVC videos without bitstream access, which can also be used for Peak Signal-to-Noise Ratio (PSNR) estimation. The results from the MPEG-2 and H.264/AVC analysis are mapped to a perceptual measure of video quality by Support Vector Regression (SVR). For validation purposes, the proposed method was tested on two databases. In both cases good performance compared with state of the art full, reduced, and no-reference VQA algorithms was achieved.**

***Index Terms*—Video Quality Assessment, No-Reference, Pixel-Based, H.264/AVC, Video Codec Analysis, PSNR estimation**

I. Introduction

VIDEO is delivered over data networks in large amounts every day, and two-way video communication has become common worldwide. Applications such as Internet Protocol television (IPTV), Video-On-Demand (VOD), commercial and home video surveillance, and video calls are widely popular. One goal for the service providers of such applications is to deliver the video in a quality that satisfies or even impresses the user. This goal leads to the need of video quality estimation. The requirements and restrictions for VQA depend on the application and perhaps also a specific use. Therefore, different methods for VQA are necessary.

VQA can be divided into the following three main categories: Full-Reference (FR), Reduced-Reference (RR) and No-Reference (NR) quality assessment. In the case of FR, the original video is accessible and the degraded version can be compared to it. In the NR scenario, the original video is not available, and therefore the quality must be estimated by solely analyzing the degraded video. The RR scenario is somewhere between these two, i.e. only some information about the original video is available for the quality estimation. NR VQA can further be divided into Pixel-Based (PB) and Bitstream-Based (BB) methods. In PB methods the analysis is based on the pixels of the decoded video (either in the spatial domain and/or other domains) without access to the bitstream, while BB methods extract parameters directly from the encoded bitstream. There are also methods that combine the two. For a more in-depth introduction to VQA see [1].

NR VQA methods are very useful since no additional data is transmitted along with the video signal. Thus, the algorithms can be carried out solely at the receiving end and without affecting the encoding or the amount of transmitted data. Video coding is normally applied to a video signal before transmission to limit the amount of transmitted data, making NR VQA methods that can evaluate videos with coding artifacts very relevant in the NR scenario. Another type of artifacts which can be encountered in videos used for transmission is channel losses. This type of artifacts are beyond the scope of this paper, but interested readers are referred to [2]–[4].

In this paper, we focus on NR PB VQA using analysis of the video encoding. Related work include analysis of the quantization in video codecs, such as MPEG-2 and H.264/AVC, and NR PSNR estimation of such videos. In [5], the authors presented a method for PSNR estimation for MPEG-2 videos using information from the bitstream. A similar approach for H.264/AVC bitstreams was proposed in [6]. The authors of [7] improved upon this method by refining the estimation of the PSNR. They also presented a metric that correlates well with perceptual opinion scores. In [8], the authors presented an H.264/AVC NR PSNR estimation method that also takes the effect of the deblocking filter into account. Unlike all of these BB methods, the authors of [9] presented a PB method for estimation of the QP parameter and motion vectors in H.264/AVC videos, but only for H.264/AVC videos where the in-loop deblocking filter has been disabled. We have previously published methods for analyzing I-frames in MPEG-2 videos and initial work on H.264/AVC [10], [11].

In this work, we present a VQA method which for H.264/AVC uses analysis of the quantization in videos with intra prediction and deblocking enabled in the encoding. With this novel method we are able to estimate important information about the video stream, such as the position of the I-frames, detection of 8x8 prediction, the quantization parameters, and the PSNR. The idea is to mimic the encoder by performing the intra-prediction to get an estimate of the transform coefficients without accessing the bitstream. We also briefly consider initial analysis of HEVC videos. The results

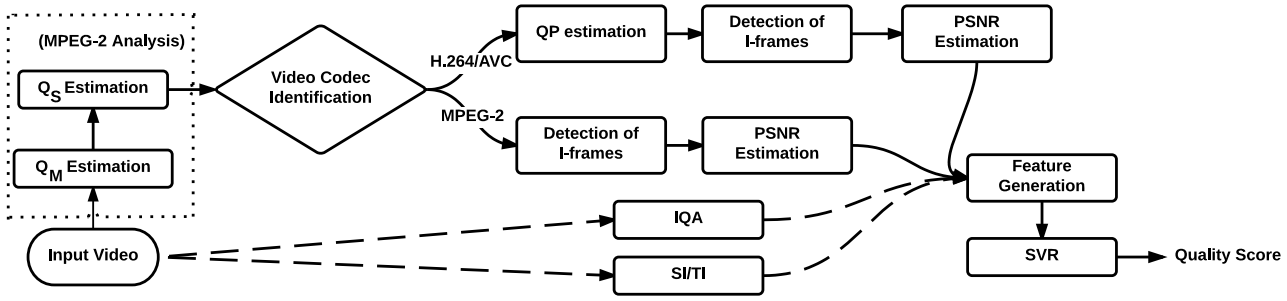

Fig. 1. Overview of the proposed NR PB VQA method. After video codec identification the corresponding codec analysis is carried out. Calculating the IQA and SI/TI features are optional.

of the analysis of the decoded videos are afterwards mapped to a video quality score using machine learning.

We consider the VQA problem in a setting where only decoded videos are available, i.e. the reference signal is not available, and we assume that the bitstream is inaccessible. Furthermore, the video codec used for encoding is regarded as unknown, but is assumed to be either MPEG-2 or H.264/AVC and further we assume a finite GOP size. Our NR PB VQA approach is useful when the video bitstream might be encrypted, e.g. an encrypted broadcast is received by a set-top box that outputs the decoded video.

The architecture of the method is illustrated in Fig. 1. First, the video codec used for encoding is detected using initial MPEG-2 analysis. Second, features are built by estimation of selected video codec parameters along with other features, either derived from an Image Quality Assessment (IQA) method, such as BRISQUE [12], applied to each frame, or based on the spatial perceptual information measure (SI) and temporal perceptual information measure (TI) [13]. Finally, the features are mapped into a quality score, using a mapping obtained by training a Support Vector Regression (SVR) [14] with video sequences that have known subjective quality scores.

The main contribution of this paper is a feature-based NR VQA method using codec analysis. To achieve this we present: a model of the distribution of the H.264/AVC transform coefficients after deblocking filtering; a novel PB H.264/AVC analysis method to estimate the quantization step, GOP size and PSNR; and a methodology to go from the feature space of the video codec analysis features to a quality score.

The paper is organized as follows. In Section II, we present our analysis of MPEG-2. In Section III, we present a model for the distribution of the transform coefficients in H.246/AVC videos. Based on this model, we present the analysis of H.264/AVC videos in Section IV, and in Section VI-F we briefly consider HEVC videos. In Section V, we explain how a modified version of the IQA method BRISQUE [12] is applied on videos, and how the features from the video codec analysis are generated and used as input to machine learning. The results of our experiments are reported in Section VI, where the performance of our method is also evaluated.

II. MPEG-2 Analysis

We use PB codec analysis to produce features relevant to the quality of the video, since we assume there is no access to the bitstream. In this Section, we present an overview of the MPEG-2 analysis [10] used for this purpose, whereas our novel H.264/AVC analysis is presented in Sections III-IV. The PB MPEG-2 I-frame analysis has been described in detail in [10], so here we only describe the overall idea of the method. The MPEG-2 codec analysis can also be used for identifying a stream as either MPEG-2 or H.264/AVC [10] as outlined at the end of this Section.

In MPEG-2, the macroblock (MB) has a size of 16×16 pixels, divided into four 8×8 DCT blocks. Two quantization parameters for the transformed coefficients are defined in the MPEG-2 standard. One is the quantization matrix Q_M that defines the quantization for each frequency of the DCT coefficients in a DCT block with indices (u, v). This matrix is the same for all DCT blocks in a frame. The second is the quantization scale parameter $Q_S(i, j)$ which is a single value that can vary from MB to MB, but stays the same for all four DCT blocks inside a single MB. Here, (i, j) is the index for the MB, but is omitted in the rest of the paper. The reconstructed AC DCT coefficients are given by:

$$\left|\tilde{\Upsilon}(u,v)\right| = \left\lfloor \frac{|\Upsilon(u,v)| \cdot Q_M(u,v) \cdot Q_S}{16} \right\rfloor, \qquad (1)$$

where $\Upsilon(u, v)$ is the received quantized values of the luminance and $\lfloor\rfloor$ denotes the floor function. Since $\Upsilon(u, v)$, $Q_M(u, v)$, and Q_S are all integers, $\tilde{\Upsilon}(u, v)$ should in principle also be an integer multiple of the quantization step: $\Delta(u, v) = 16^{-1}Q_M(u, v)Q_S$.

We assume that the number of potential Q_M parameters is smaller than the number of potential Q_S parameters. Therefore, we start the analysis by estimating the Q_M parameter. For each MB, candidate Q_M, and a given Q_S, a mismatch value M_{MB} is introduced and defined as:

$$M_{MB}(Q_M, Q_S) = \sum_{(u,v)\in MB} \left| \left[\frac{\tilde{\Upsilon}(u,v)}{\Delta(u,v)} \right] - \frac{\tilde{\Upsilon}(u,v)}{\Delta(u,v)} \right|, \qquad (2)$$

where $[\,]$ denotes rounding to nearest integer. The minimum value of (2) is determined for each MB and candidate Q_M and then summed over all MBs in a frame. The candidate Q_M resulting in the lowest sum is chosen as the estimated quantization matrix $\tilde{Q}_M$. The estimation of the Q_S parameter is similar, but based on the individual MBs. For the details of this estimation and how it can be used for PSNR estimation we refer to [10].

As a measure of uncertainty of the MPEG-2 analysis the *mismatch value* for the whole frame is defined as the mean value of (2) with the estimated quantization values as input. The *mismatch value* can be used for codec validation and thereby codec identification and estimation of the I-frame positions, since frames which are not MPEG-2 I-frames will result in high uncertainty. Thus, a video where this analysis consistently produces high *mismatch values* for each frame is regarded as not being MPEG-2 encoded. Using this approach for the MPEG-2 and H.264/AVC videos in the databases described in Section VI we achieved 100% identification accuracy for both databases. In a setting where the videos are not limited to MPEG-2 and H.264/AVC, a codec identification method such as [15] could be applied instead.

III. H.264/AVC Quantization

To produce H.264/AVC codec features, we present in this Section, our model of the distribution of the transformed coefficients in H.264/AVC videos after deblocking filtering. This is necessary since the deblocking filter has a high impact on the distribution of the transform coefficients and should not be ignored in the PB case. The model is used in the PB analysis presented in Section IV.

All of our analysis is based on the luminance values of the decoded frames. In H.264/AVC the residual blocks are transformed with a 4×4 or 8×8 transform with similar energy compaction properties as DCT. The DC coefficients in the 16×16 blocks can be further transformed with an Hadamard transform. When quantizing the transform coefficients, the encoder chooses the quantization parameter QP, which can be different for each MB. The reconstructed transform coefficients are ideally at the encoder given by

$$\left|\tilde{Y}(u,v)\right| = \left\lfloor \frac{|Y(u,v)|}{q(u,v)} + 1 - \alpha \right\rfloor q(u,v), \tag{3}$$

where $Y(u,v)$ is the original quantized (and scaled) prediction residuals, α is a parameter deciding the shift of the quantization boundary, and $q(u,v)$ is the quantization step defined as Q_{step} in [16]. As stated in [9], when the QP parameter is large enough ($QP > 20$) it can be estimated by using a maximum likelihood estimator. However, if the deblocking filter is enabled, the estimation given in [9] is not very robust and is likely to fail to estimate the correct QP parameter. Instead, we use the model presented in the following subsections before estimating the QP parameter.

A. Modeling the coefficient distribution

In order to analyze the DCT coefficients, the distributions of the coefficients with different quantization parameters can be modeled [17]–[19]. The model chosen here is a mixture of Cauchy distributions motivated by e.g. [18], [19] and given by

$$f(x, qs) = \sum_{i=0}^{N} w_i p_i(x, qs, \gamma_i), \tag{4}$$

where N is the number of distributions in the mixture model for signal x, γ_i is the γ parameter in the ith Cauchy distribution, and w_i are weights. Thus, p_i is the Cauchy distribution

$$p_i(x, qs, \gamma_i) = \frac{1}{\pi} \frac{\gamma_i}{(x - i \cdot qs)^2 + \gamma_i^2}, \tag{5}$$

where qs is given by

$$qs(QP) = 0.6249 e^{0.1156 QP}. \tag{6}$$

The expression in (6) is an approximation to the relation between the quantization step and the QP parameter [16]

$$qs(QP) = qs_B(QP \bmod 6) 2^{\lfloor QP/6 \rfloor}, \tag{7}$$

where qs_B is the base quantization step as detailed in [16].

Without any quantization the distribution of the coefficients can be modeled by a single Cauchy distribution with zero mean [17], [18]. Therefore, when taking the quantization into account using a mixture of Cauchy distributions, the weight of the distributions decline very rapidly with increasing distance from zero. We have also observed this empirically[1]. This gets more pronounced for higher quantization parameters since the center of the Cauchy distributions (for $i > 0$) are further away from zero. We also observed, that the higher the quantization parameter, the wider the Cauchy distributions seem to be. Based on these observations, the γ_i parameter of the Cauchy distributions can be modeled as a first order polynomial

$$\gamma_i(QP) = a_i + b_i \cdot QP. \tag{8}$$

Since the Cauchy distribution with $i = 0$ always has zero mean, it is much harder to see the effect of different quantization steps as opposed to distributions with $i > 0$. Therefore, we disregard the contribution from the Cauchy distribution with $i = 0$. Due to the rapid decline of the weights along the distance from zero, we only consider the cases where i is equal to 1 or 2. For $i = \{1, 2\}$ we empirically found $a_i = \{-3.12, -2.55\}$ and $b_i = \{0.19, 0.15\}$. The weights in the mixture also depends on the quantization scale, but were found to be in the range of $[0.75, 1]$ for w_1, while $w_2 = 1 - w_1$.

Since the QP parameter in H.264/AVC can change from MB to MB, we base our analysis on the individual MBs. Therefore, when using this information for estimating the QP parameter in a decoded video sequence, one must keep in mind the very limited number of samples available in a MB. In our testing, we found that traditional "goodness of fit" methods were ineffective to determine the distribution of coefficients. Our approach is to calculate a weighted sum of the coefficients, where we use a slightly modified mixture of two Cauchy distributions as the weights. Since the peak of the Cauchy distributions are lower for high QP parameters, without any modification we would weigh coefficients at higher quantization steps less than coefficients at lower quantization steps, but since that effect is unwanted, we use the following modified

[1]Empirical observations and results are based on the H.264/AVC videos in the LIVE database [20]. The details of the database can be seen in the beginning of Section VI. The observations are validated by the performance on the independent database denoted Lisbon (also detailed in Section VI).

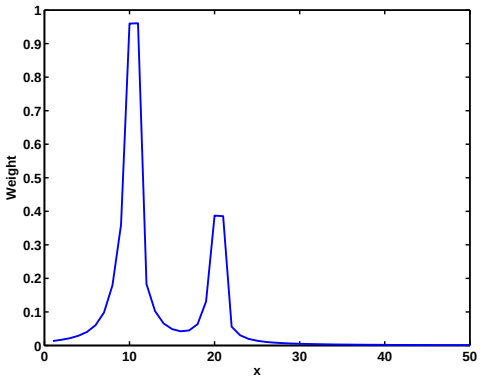

Fig. 2. The proposed weighting function $\hat{f}(x, QP)$ for $QP = 24$ as a function of the absolute values of the transform coefficients.

version of the distributions in the mixture

$$\hat{p_i}(x, QP) = \begin{cases} \max(p_i) & \text{if } x{=}\lfloor i{\cdot}qs(QP)\rfloor \\ & \text{or } x{=}\lceil i{\cdot}qs(QP)\rceil \\ p_i(x, qs(QP), \gamma_i(QP)) & \text{otherwise} \end{cases} \quad (9)$$

where $\max(p_i)$ is the maximum p_i value over all quantization steps. The mixture used as weighting can then be written as

$$\hat{f}(x, QP) = \sum_{i=1}^{2} w_i(QP)\hat{p_i}(x, qs(QP), \gamma_i(QP)). \quad (10)$$

where $w_i(QP)$ are the weight parameters of the mixture model and found empirically to be in the range of $[0.75, 1]$ for $w_1(QP)$, while $w_2(QP) = 1 - w_1(QP)$. The resulting weighting function for $QP = 24$ where $qs(24) = 10.02$ can be seen in Fig. 2. This is used in the MB analysis as explained in the next Section.

IV. H.264/AVC I-frame Analysis

In this Section, we use our model of the transformed coefficients in H.264/AVC I-frames from Section III to estimate the quantization step and thereby the Mean Squared Error (MSE) and PSNR of a H.264/AVC encoded video. The goal of the analysis is not the estimations themselves, but rather to build useful features, which can be used in the prediction of a quality score that correlates well with subjective opinions. If there exist prior knowledge about the positions of the I-frames, the analysis described in this Section can be limited to those frames, otherwise the analysis needs to be carried out on every frame. The latter has been done in our experiments. We do not design specific codec analysis for P- or B-frames since that would increase the complexity of our method substantially due to the computational complexity of H.264/AVC motion estimation. Consequently, our model is only designed for videos with finite GOP size, since the information extracted from a single I-frame in the start of a video with infinite GOP size is not necessarily characteristic for the rest of that video.

For each frame in a video to be analyzed, we carry out standard H.264/AVC intra prediction for 4x4, 8x8 and 16x16 blocks using the Sum of Absolute Differences (SAD) as the criteria to find the best match. This is done for each MB and results in three residual frames; one for each prediction block size. For each original MB in the frame, there is only one of the respective MBs in the estimated residual frames with the correct prediction block size, but which one is unknown. The transform coefficients of each residual frame is analyzed as outlined in Sections IV-A to IV-B. The statistics of the results are gathered as codec information as described in Section IV-B. This information is used in Section IV-C to estimate the positions of the I-frames and in Section IV-E to estimate the PSNR for each estimated I-frame.

A. MB analysis

In each residual frame we calculate a *response* for each MB and $QP \in [21, \ldots, 51]$. The value of the *response* reflects how well the actual distribution of coefficients in a MB matches our expectations for different QP values. We limit the calculation to MBs where the maximum coefficient C_{max} is greater than a threshold τ_1 and the number of non-zero coefficients I_{NZ} is greater than a threshold τ_2, i.e. $C_{max} \geq \tau_1$ and $I_{NZ} > \tau_2$. This is motivated by the fact that in MBs where most coefficients are of very low values the quantization step cannot be identified due to noise. The *response* is based on (10)

$$R(QP) = qs(QP) \sum_{j=0}^{M_x} H_j \hat{f}(x_j, QP), \quad (11)$$

where M_x equals the maximum of the quantized values x_j, and H_j is the number of quantized values equal to x_j. In our implementation, we set $\tau_1 = 49, \tau_2 = 9$.

The reason for multiplying with qs is that $R(QP)$ otherwise seems to be an exponential decaying function, due to the fact that the original coefficient value can be modeled as a Cauchy distribution with zero mean. Even after including the $qs(QP)$ part in (11), the function might still have an overall linear decreasing trend, which can be due to the effect of the deblocking filter. Since we are interested in the strongest relative response, we fit the $R(QP)$ function with a polynomial $G(QP)$ of degree 1:

$$G(QP) = aQP + b, \quad (12)$$

where the coefficients a, b are found by a least squares fit of $G(QP)$ to $R(QP)$. Then we can calculate the estimation of the QP value for the MB by

$$q = \underset{QP}{\operatorname{argmax}} \left(R(QP) - G(QP)\right). \quad (13)$$

Thus, for the whole residual frame we get a matrix of QP predictions denoted $\hat{QP}^i_{MB}$, where i is an index of the prediction size and each element is found by (13).

B. QP estimation

For each frame we collect statistics based on the MB analysis, which can later be used in the machine learning part. As our estimation of the QP parameter for each residual frame, we simply select the QP value which was chosen the most times by the MB estimation, i.e. it is defined by

$$QP^i_{est} = \underset{QP}{\operatorname{argmax}} \sum \sum \theta(QP_{can}), \quad (14)$$

where $\theta(QP_{can})$ is an indicator matrix where the elements are equal to 1 if the corresponding elements in $\hat{QP}^i_{MB}$ are equal to the candidate QP_{can} and 0 otherwise.

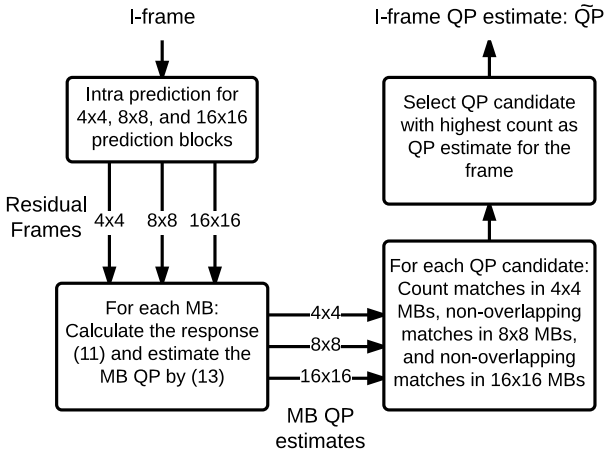

Fig. 3. Block diagram of the I-frame QP estimation

Based on the three QP^i_{est} values we calculate an overall estimation of the QP parameter for the frame. For each value we count for how many MBs this value is output by the estimation performed in the 4x4 residual frame. For all non-matching MB we check whether there is a match in the corresponding 8x8 MB and finally in the 16x16 MB. Then the estimated QP parameter with highest total number of matching MBs is chosen as the QP estimation for the frame $\widetilde{QP}$. In this way, the 4x4 residual has higher priority than the 8x8 and 16x16 residual frames and the 8x8 residual frame has higher priority than the 16x16 residual frame, which is motivated by the observation that the reliability of the estimations for smaller prediction blocks are usually better than for larger prediction blocks. A summary for this estimation is shown in the block diagram in Fig. 3.

To build features for the machine learning, besides the overall estimated QP parameter for the frame, we produce statistics of our analysis when there is at least one MB in a residual frame with an estimation of the QP i.e. a MB satisfying the constraint imposed by the thresholds τ_1, τ_2. Let $N^i_{QP}, i \in \{4, 8, 16\}$ be the number of MBs in a residual frame, where QP^i_{est} from the MB estimation matches the selected overall frame QP estimation value. Let N^i_{tot} be the total number of MBs for which the response (11) was calculated in the residual frame with index i. Let N_{frame} be the total number of MBs in the frame. If N^i_{tot} is above 9, we collect the following statistics for the corresponding residual frame: QP^i_{est} (the estimated QP), P^i_{con} (the fraction N^i_{QP}/N^i_{tot} which is used as a measure of confidence), P^i_{tot} (the fraction N^i_{tot}/N_{frame}), and P^i_0 (the fraction of zero residual MBs).

Confusion matrices for the real and estimated QP values for the I-frames in the LIVE and Lisbon database are depicted in Figs. 4 and 5, respectively. Test sequence details can be found in Section VI and in [7], [20]. As can be seen in the figures the estimation of the frame QP values for the Lisbon database seems to be worse than for LIVE. This is due to the fact that the range of the QP values used in the Lisbon database are much larger than in the LIVE database. If the RMSE for prediction in the Lisbon database is calculated using only frames with original QP values in the same range as for LIVE, the RMSE is equal to 0.62.

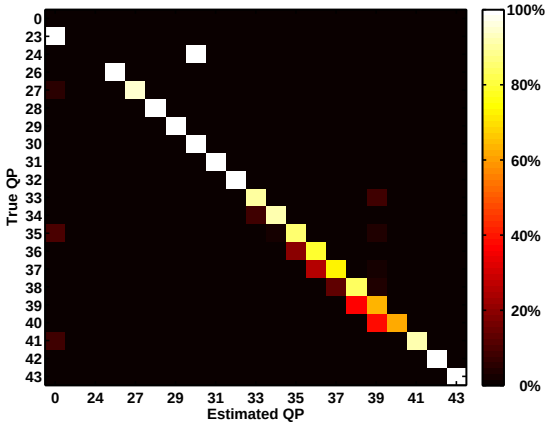

Fig. 4. I-frame QP confusion matrix for the LIVE database. The estimated QP is equal to 0 when $N^i_{tot} < 9$, $\forall i$, i.e. we do not have enough data for that frame (which is the case for 2% of the I-frames). For the rest of the estimated QP values the RMSE is equal to 0.77. (The reason for the apparently large error at original $QP = 24$ is due to the fact that there is only two frames with this particular QP in the whole dataset and they are both wrongly estimated.)

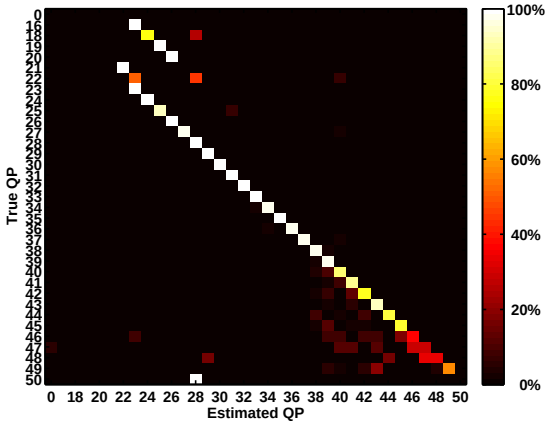

Fig. 5. I-frame QP confusion matrix for the Lisbon database. The estimated QP is equal to 0 when $N_{tot} < 9$, $\forall i$, i.e. we do not have enough data for that frame (which is the case for 0.1% of the I-frames). For the rest of the estimated QP values the RMSE is equal to 2.2. (Since the proposed method only considers $QP \in [21, \ldots, 51]$, all the original QP parameters outside this range is wrongly estimated.)

In this paper, we assume that the QP parameter is constant in a frame, but our analysis has been performed with variable QP parameters inside a frame in mind. Therefore, we produce QP estimations for each MB instead of just one estimate for the whole frame. In this paper, the final QP prediction for the whole frame is given by (14). If variable QP inside a frame should be taken into account, our method could be adjusted such that the overall QP estimate for a frame would be a weighted average of the QP parameters in the frame.

C. GOP size estimation

In this work, we assume that the GOP size for the video sequences is fixed. If the GOP size varies in a video, alternative methods such as [15] is needed to estimate the position of the I-frames. Our approach to estimate the position of the I-frames from the decoded video pixels is based on the P^i_{con} statistics

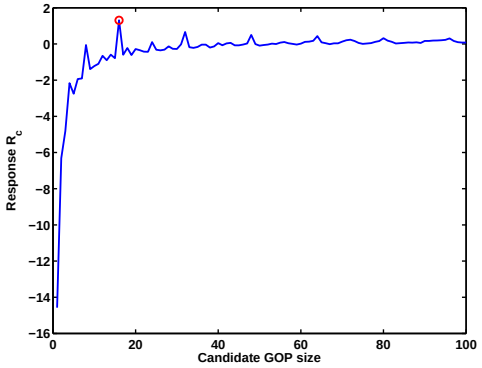

Fig. 6. Illustration of the response R_c as function of candidate GOP sizes for the "Pedestrian Area" video at highest bitrate from the LIVE database. The value of the true GOP size is marked by a red circle. This value coincides with the input value giving the maximum response R_c value for this example.

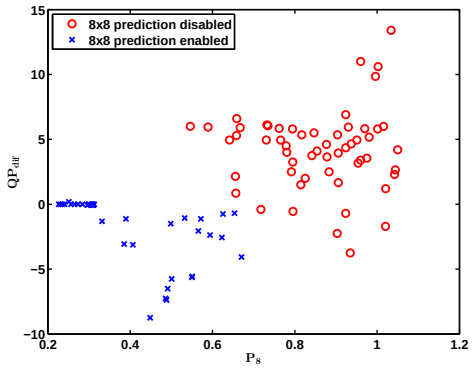

Fig. 7. Detection of 8x8 prediction using the QP_{dif} (18) and P_8 (19) values.

(Section IV-B), which are used to produce a vector $\mathbf{c}$ where the value of an element $\mathbf{c}_j$, represents the reliability or confidence of the analysis for the corresponding frame

$$\mathbf{c}_j = \max_{i\in\{4,8,16\}} P^i_{con}(j), \tag{15}$$

where j is the index of the frames. The actual values in of the elements in $\mathbf{c}$ also depend on the content of the analyzed video. To reduce this dependency, we subtract the mean and the standard deviation (std) of the vector from the values

$$\tilde{\mathbf{c}}_j = \mathbf{c}_j - \text{mean}(\mathbf{c}) - \text{std}(\mathbf{c}). \tag{16}$$

Then, for each candidate GOP size s, we create a filter vector $\mathbf{v}^s$ where the elements at positions equal to $ks+1$ for all $k \in \mathbb{Z}^*$ are set to 1 and all other elements are set to 0. Thereafter, a response denoted $R_c(s)$ is calculated by summing the element-wise product of $\tilde{\mathbf{c}}$ and the filter vector,

$$R_c(s) = \sum_{j=1}^{N} \tilde{\mathbf{c}}_j \mathbf{v}^s_j, \tag{17}$$

where j is the index of the elements in the vectors and N is the number of analyzed frames. Finally, the candidate GOP size s with the greatest value of $R_c(s)$ is selected as the estimation of the GOP size.

An example for a sample video can be seen in Fig. 6. In this paper, we assume that the start of the analyzed video is also the beginning of a GOP. This assumption can be relaxed by introducing an integer offset j, so that every element at positions equal to $ks+j$ for any $k \in \mathbb{Z}^*$ is set to 1 in the filter vector. If the GOP size varies, this method cannot be used and an alternative approach such as the one presented in [15] would be needed.

In our experiments, we used candidate GOP sizes in the interval $[1, 2, \ldots, 100]$ and got perfect accuracy in the prediction for both databases used in the testing when not using the offset parameter for the filter vector. When using the offset, we still get perfect accuracy for the LIVE database but for 6 out of the 56 videos in the Lisbon database the GOP becomes wrongly estimated. Due to this generally high accuracy of I-frame detection, we assume in our experiments that the position of the I-frames is known when we are estimating the quality.

D. Detection of 8x8 prediction

Our H.264/AVC analysis can also be used to detect whether prediction with a block size of 8x8 has been used during encoding. This is based on the estimated QP values for the different block sizes used for prediction, and the confidence measures P^8_{con} and the percentage of used MBs P^8_{tot} for 8x8 prediction residual frames in the I-frames versus the corresponding values for the inter-predicted frames.

We use the following values denoted QP_{dif} and P_8 based on the QP^i_{est} values from (14) and the P^i_{con} values from Section IV-B to detect whether 8x8 prediction was used in the H.264/AVC encoding of the videos.

$$\begin{aligned} QP_{dif} &= \frac{1}{N_I}\sum_{k\in I} |QP^4_{est}(k) - QP^8_{est}(k)| \\ &\quad - \frac{1}{N_I}\sum_{k\in I} |QP^4_{est}(k) - QP^{16}_{est}(k)|, \end{aligned} \tag{18}$$

$$P_8 = \frac{\sum_{k\notin I} P^8_{con}(k) P^8_{tot}(k)}{\sum_{k\in I} P^8_{con}(k) P^8_{tot}(k)}, \tag{19}$$

where I is the set of I-frame positions and N_I is the number of I-frames in the video. These values for the videos in the LIVE and Lisbon database are depicted in Fig. 7. As evident from the figure, the videos with 8x8 prediction enabled can be separated from those without in every case.

E. PSNR estimation

Bitstream Based estimation of the PSNR for H.264/AVC encoded videos have been investigated in [6]–[8]. In this subsection, we present and evaluate our Pixel Based PSNR estimate for I-frames in H.264/AVC videos. It should be noted, that the PSNR estimate is not a goal in itself here, but it is used as an input to the machine learning algorithm. As detailed in [7], the overall MSE can be calculated by the sum of MSE values ε^2_k of the transform subbands. If the distribution of the original transform coefficient data is known $\varepsilon^2_k(MB)$ at the kth coefficient position in a single MB can be estimated by using the quantized value X_k [7]:

$$\varepsilon^2_k(MB) \approx \frac{\int_{a_k}^{b_k} f_X(x)(X_k(MB) - x)^2 dx}{\int_{a_k}^{b_k} f_X(x) dx}, \tag{20}$$

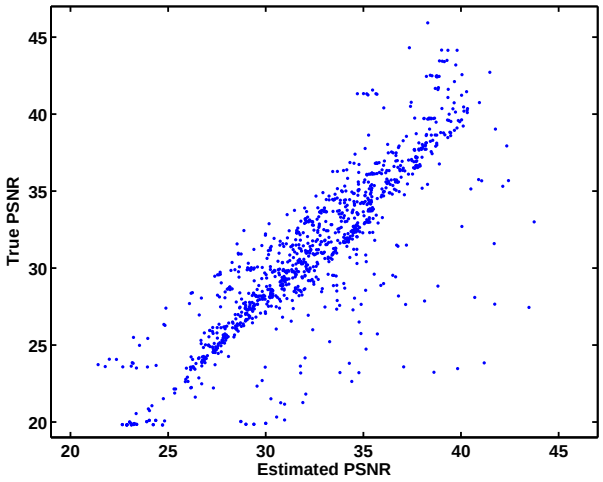

Fig. 8. Scatter plot of the PSNR estimation for analyzed frames in the Lisbon database. SROCC: 0.87.

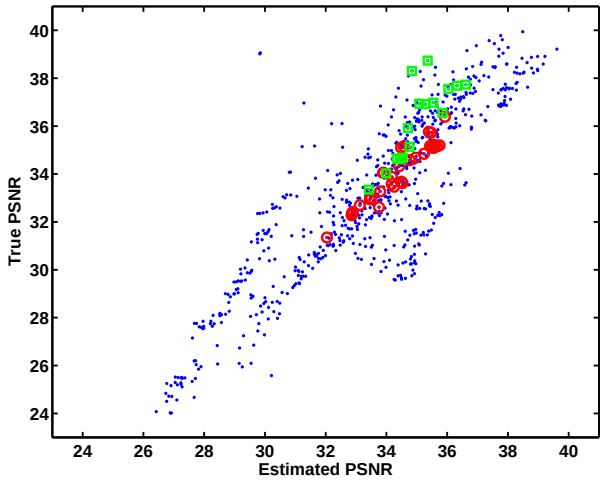

Fig. 9. Scatter plot of the PSNR estimation for analyzed frames in the LIVE database (□ and ○ marks I-frames of two selected sequences). SROCC: 0.82.

where $f_X(x)$ is the original distribution of the coefficients at the corresponding transform block position.

For the whole frame we can approximate the overall subband MSE ε_k^2 at the kth coefficient position by averaging over all quantized values in the corresponding transform positions in the frame. This can be expressed as a weighted summation of each of the quantization intervals indexed by j (which are defined by the QP value):

$$\varepsilon_k^2 \approx \sum_j P_k(X_j) \frac{\int_{a_j}^{b_j} f_X(x)(X_j - x)^2 dx}{\int_{a_j}^{b_j} f_X(x) dx}, \tag{21}$$

where $P_k(X_j)$ is defined as the ratio of coefficients in the transform position k, which is inside the jth quantization interval belonging to X_j (which is bounded by a_j and b_j).

To model the DCT-like coefficient distribution, we again use a Cauchy distribution motivated by [18], [19]:

$$f_X(x, \gamma) = \frac{1}{\pi} \frac{\gamma}{x^2 + \gamma^2}. \tag{22}$$

The shape parameter γ_k, for coefficient k, is estimated by the percentage P_0 of coefficients with a magnitude lower than $\alpha \cdot qs$ in the reconstructed coefficients for the given coefficient position, since as shown in [6]:

$$\begin{aligned} P_0 &= 2 \int_0^{\alpha \cdot qs} f_X(x, \gamma_k) \\ P_0 &= \frac{2}{\pi} \arctan \frac{\alpha \cdot qs}{\gamma_k}, \end{aligned} \tag{23}$$

we get [6]

$$\gamma_k = \frac{\alpha \cdot qs}{\tan \frac{\pi P_0}{2}}. \tag{24}$$

We can use (21) to calculate an approximation for the subband MSE. Since we do not access the bitstream, the true quantized coefficients X_j defined by the frame QP value are not available. Therefore, we use the estimated quantization step size and quantize the estimated reconstructed coefficients $\hat{X}_j$ retrieved from the estimated residual frame instead. All MBs in the estimated residual frame, satisfying that the estimated QP value in the MB is equal to the overall estimated QP for that frame, is used to calculate the subband MSE for each coefficient in the transform block. In this analysis for the sake of simplicity, we disregard MBs with estimated transform block size of 8x8 and only consider estimated prediction blocks of 4x4 and 16x16, for which the transform block size is 4x4. Using (21) and (22) the subband MSE is calculated by the contributions from the quantization interval where $\hat{X}_j = 0$ and the intervals where $\hat{X}_j \neq 0$. This is similar to the approach in [6] where a Laplacian distribution is assumed. If $\hat{X}_j = 0$ the contribution is calculated by:

$$\hat{\varepsilon}_{k,j}^2 = P_k(\hat{X}_j) \left(\alpha \gamma_k qs - \gamma_k^2 \arctan \frac{\alpha \cdot qs}{\gamma_k} \right), \tag{25}$$

where $P_k(\hat{X}_j)$ is the ratio of reconstructed coefficients in the transform position k which is inside the jth quantization interval belonging to $\hat{X}_j$. If $\hat{X}_j \neq 0$ the contribution is:

$$\begin{aligned} \hat{\varepsilon}_{k,j}^2 &= P_k(\hat{X}_j) \gamma_k qs \\ &+ P_k(\hat{X}_j)(\hat{X}_j^2 + \gamma_k^2) \arctan \frac{\hat{X}_j + \alpha qs}{\gamma_k} \\ &- P_k(\hat{X}_j)(\hat{X}_j^2 + \gamma_k^2) \arctan \frac{\hat{X}_j - (1-\alpha) qs}{\gamma_k} \\ &+ P_k(\hat{X}_j) \gamma_k \hat{X}_j \log \left(\frac{(\hat{X}_j - (1-\alpha)qs)^2 + \gamma_k^2}{(\hat{X}_j + \alpha qs)^2 + \gamma_k^2} \right). \end{aligned} \tag{26}$$

Finally we can get an estimation of the MSE (MSE_{est}) by summing over the quantization intervals and coefficient positions, which can be used to calculate an estimate of the PSNR ($PSNR_{est}$) for the frame by

$$MSE_{est} = \frac{1}{16} \sum_{k=1}^{16} \sum_j \hat{\varepsilon}_{k,j}^2 \tag{27}$$

$$PSNR_{est} = 10 \log_{10} \frac{255^2}{MSE_{est}}. \tag{28}$$

The PSNR estimation was compared to the actual PSNR of the I-frames where an estimation of the QP value was calculated.

That is, for I-frames where there were at least one estimated MB satisfying the constraint imposed by the thresholds τ_1, τ_2 used during the MB analysis. The data is taken from the 200 first frames of all H.264/AVC videos in the two databases used for testing and the results are depicted in Figs. 8 and 9.

To demonstrate that PSNR is not sufficient to evaluate the quality across different contents, the I-frames from two sequences with different content have been highlighted in Fig. 9. The value of 100-DMOS for the sequence highlighted with green squares is 45.1 (lower quality), while for the sequence highlighted by red circles it is 59.2 (higher quality). Using our VQA approach with the features denoted F10 as explained in Section VI the means of the predictions across our cross validation folds are 50.2 and 58.1, respectively. Thus, despite generally higher PSNR values for the sequence marked with green squares in Fig. 9 the proposed method correctly ranks the sequence marked with red circles higher in terms of quality.

V. Video features and Machine Learning

To achieve video quality assessment, we apply machine learning based on codec and image features. For image and video quality assessment, many machine learning methods such as neural networks have been used, e.g. [21]–[24]. Recently, the Support Vector Machine (SVM) used for regression, known as Support Vector Regression (SVR), seems to gain popularity in the fields of IQA and VQA due to generally obtaining high performance [12], [25]–[29]. In this Section, we describe how we use as inputs, the codec analysis features obtained as briefly described in Section II for MPEG-2 and as described in Section IV for H.264/AVC. We also introduce new features based on an IQA method and on the spatial perceptual information measure (SI) and temporal perceptual information measure (TI) [13] as complementary features. The SVR algorithm is also briefly described in this Section.

A. MPEG-2 Analysis Features

For MPEG-2 the codec analysis features are based on the PSNR estimates [10] for the I-frames. Using the temporal pooling in [30], we divide the PSNR estimates into a high cluster C_H (i.e. a set of high PSNR values) and a low cluster C_L (i.e. a set of low PSNR values) using k-means clustering (with $k = 2$). Thereafter, we calculate a weighted average

$$\mu_w = \frac{\sum_{i \in C_L} PSNR_i + w \sum_{i \in C_H} PSNR_i}{|C_L| + w\,|C_H|}, \tag{29}$$

where the weight is defined by the average of the high and low cluster, denoted μ_H and μ_L, respectively,

$$w = \left(1 - \frac{\mu_L}{\mu_H}\right)^2. \tag{30}$$

This temporal pooling is motivated by the fact that high and low quality segments in a video are not equally important for the perceived quality. We use μ_w, μ_L, μ_H, and w as features. We include the average estimated PSNR, and the standard deviation of the estimated PSNR, the standard deviation of the estimated PSNR in the low cluster and the standard deviation of the estimated PSNR in the high cluster. The maximum and minimum estimated PSNR values is also included along with the absolute difference of the two values. Furthermore, we use the so-called *mismatch* values and the estimated Q_S values as mentioned in Section II and detailed in [10]. In each frame, we take the standard deviation and the mean of these values over each frame and as temporal pooling we again use the mean and standard deviation resulting in 8 more features. In total, 19 features are included from the MPEG-2 analysis.

B. H.264 Analysis Features

For H.264/AVC we use the average and standard deviation of the feature values over the I-frames as temporal pooling. The features are the means of the QP^i_{est}, the standard deviation of QP^4_{est}, the mean of P^4_{con}, the mean and standard deviation of the sum of the three P^i_{tot}, the mean of $\tilde{QP}$ and the mean and standard deviation of $PSNR_{est}$. We also calculate a weighted QP estimate wQP for each I-frame indexed by k

$$wQP(k) = \frac{\sum_i QP^i_{est}(k) P^i_{con}(k)}{\sum_i P^i_{con}(k)}. \tag{31}$$

We use the minimum, maximum, the mean and the standard deviation of wQP as features and we also calculate a weighted single QP estimation feature over the I-frames,

$$\tilde{w}QP = \frac{\sum_k \sum_i QP^i_{est}(k) P^i_{con}(k)}{\sum_k \sum_i P^i_{con}(k)}. \tag{32}$$

Thus, a total of 15 features is calculated based on the H.264/AVC analysis. The estimated PSNR features are based on (28), while the rest of the features are based on the statistics given in Section IV-B.

C. Image Quality Assessment Features

To perform PB NR IQA, we have chosen to modify BRISQUE [12], that has shown good performance for images with various types of noise. BRISQUE is based on natural scene statistics and calculates a set of features for an image. Thus, in our implementation we can use it to get a set of features for each frame. These features can then be pooled into a single feature set for the whole video sequence, and they can be used along with our video codec features as input to the SVR. The features in BRISQUE are all based on the so-called Mean Subtracted Contrast Normalized (MSCN) coefficients which are calculated by

$$\hat{I}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + K}, \tag{33}$$

where I is the luminance of an image, (i, j) are the spatial indices of pixels, and $K = 1$ is a constant preventing numerical instabilities. $\mu(i, j)$ and $\sigma(i, j)$ are the weighted mean and the weighted standard deviation of local luminance values as detailed in [12]. Furthermore, in BRISQUE a single downscaling with a factor of 2 is performed and used to calculate the MSCN coefficients on this scale.

In the presentation of BRISQUE [12], it is shown how the distribution of the MSCN coefficients are altered when an

image is distorted by different types of noise. Our experiments have shown that this alteration seems to be less severe with still frames from MPEG-2 and H.264/AVC videos. Also, frames containing areas with large constant surfaces will lead to a lot of zero MSCN coefficients, altering the distribution. If such an area arises from compression, it seems to be less severe in the downscaled version, contrary to the case where the area is also present in the original video.

Based on these observations, we alter the BRISQUE analysis (with 36 features) to video still frames by using slightly different and fewer features. For the MSCN coefficients we calculate the shape parameter and the standard deviation of a generalized Gaussian distribution fitted to the MSCN coefficients, where $|\hat{I}(i,j)| > \epsilon$, ϵ being a small value used to avoid MSCN coefficients close to zero (in BRISQUE $\epsilon = 0$). For the diagonal pairs defined in [12], we only calculate the variance of coefficients above and below zero, leaving out the shape and the mean parameter also used in BRISQUE. Thus for each scale of the image, we get 10 features and since we only downscale once, the total set consists of 20 features.

If there is a lot of small MSCN coefficients on one scale, but not on the other, this could be a sign of compression artifacts e.g. having constant luminance values inside a MB due to compression. Therefore, we also introduce a feature expressing the fraction of small MSCN coefficients for the two scales

$$R_Z = \frac{M_\epsilon^d}{M^d} \cdot \frac{M}{M_\epsilon}, \tag{34}$$

where M and M^d are the total number of MSCN coefficients in the original resolution and in the downscaled version respectively, and M_ϵ and M_ϵ^d are the corresponding number of coefficients where $|\hat{I}(i,j)| < \epsilon$. Thus, in total we get 21 features per frame. To get a single feature vector for the whole video, we take the average and the standard deviations of the frame features for selected frames (e.g. all frames or only inter frames), in total giving 42 IQA features for each video. The feature set can be used alone or together with features based on the video codec analysis. In the latter case, only the 21 standard deviation values of the features are used.

D. Spatial and Temporal Information

To get information about the spatial and temporal complexity in the videos we have used the spatial perceptual information measure (SI) and temporal perceptual information measure (TI) from [13] on the distorted videos. Since they are calculated on the distorted videos, they will depend on the amount of distortion. Nevertheless, the measures still contain information about the spatial and temporal complexity of the videos, which will be useful in our machine learning approach. Instead of using the maximum value over time, we instead use the average and standard deviation of the SI/TI values of each frame as additional features. These features are only used when the IQA features are not included.

E. Support Vector Regression

To map our features to a quality score we use SVR, namely the ϵ-SVR method implemented in LIBSVM [31]. The aim is to find a function of the feature vector $\mathbf{v}$

TABLE I
CHARACTERISTICS OF VIDEO TEST SEQUENCES

	H.264/AVC		MPEG-2	
	LIVE	**Lisbon**	**LIVE**	**Lisbon**
Original videos	10	12	10	8
Bitrate levels	4	4-6	4	4
Resolution	768x432	352x288	768x432	352x288
Bitrates [Kbps]	200-5000	32-2048	700-8000	128-4096
Framerate	25/50	25/30	25/50	25/30
Duration [s]	10	8.9-10	10	8.9-10
8x8 Prediction	Yes	No	-	-
GOP Size	16	15	15	15
GOP Structure	IPPPP	IBBPBBP	IBBPBBP	IBBPBBP

$$f(\mathbf{v}) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(\mathbf{v_i}, \mathbf{v}) + \beta_0, \tag{35}$$

where $(\alpha_i^* - \alpha_i)$ are the solution values, $K(\mathbf{v_i}, \mathbf{v})$ is the kernel function and β_0 is an offset. The feature vectors $\mathbf{v_i}$ where $(\alpha_i^* - \alpha_i)$ is non-zero are the so-called support vectors. Interested readers are referred to [14]. We use the radial basis function as the kernel function

$$K(\mathbf{v_i}, \mathbf{v}) = e^{-\omega ||\mathbf{v} - \mathbf{v_i}||^2}. \tag{36}$$

When training the model, we search for the optimal values of three parameters, the cost and the ϵ parameter in the SVR formulation and the ω parameter in the radial basis function, in a 3-dimensional grid search.

We use cross-validation to avoid overfitting. In every test, we left out two source contents coded at all different bitrates for computation of the test error. We performed this test for each possible content independent split into training and test sets of videos. This testing approach is more comprehensive than traditional k-fold cross-validation. In the results, we report the median, average and standard deviations of these tests.

VI. RESULTS

In our evaluation, decoded MPEG-2 and H.264/AVC video streams from the LIVE database [20] and from the database in [7] (denoted as Lisbon) were used. The Lisbon database only contains videos with distortion solely due to compression. The LIVE database also includes videos with transmission errors. We only use a subset of the LIVE database, i.e. the videos that only contain compression artifacts. The characteristics of the test sequences used are summarized in Table I. Both datasets were encoded with the JM reference software [32] using the standard rate control algorithm (version 12.3 for LIVE and version 12.4 for Lisbon). The opinion scores for the two databases are reported differently, so we rescale the MOS values for Lisbon to the interval $[0, 100]$ and for LIVE, we take 100-DMOS as the measure of quality.

To evaluate the performance of our NR PB VQA (Section V) without access to the bitstream, we used different data and feature sets as seen in Tables II-IV. The detailed characteristics of the data and feature sets are the following: (F1) Mixed H.264/AVC and MPEG-2 streams that were analyzed only with IQA (denoted Mixed). Training the SVR with IQA features on the Mixed dataset, but testing on (F2) H.264/AVC and (F3) MPEG-2, separately. Training *and* testing on (F4)

TABLE II
SROCC CROSS-VALIDATION RESULTS

	LIVE			Lisbon		
	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ
Only IQA features						
(F1) Mixed	0.58	0.48	0.29	0.84	0.78	0.14
(F2) H.264	0.64	0.47	0.46	0.82	0.76	0.19
(F3) MPEG2	0.60	0.60	0.24	0.9	0.83	0.15
IQA with separation						
(F4) H.264	0.67	0.53	0.35	0.86	0.79	0.19
(F5) MPEG2	0.80	0.71	0.21	0.83	0.77	0.18
IQA + codec features						
(F6) H.264	0.86	0.78	0.23	**0.96**	0.94	0.057
(F7) MPEG2	**0.93**	0.84	0.17	**0.98**	0.97	0.024
Only codec features						
(F8) H.264	0.88	0.78	0.24	0.95	0.94	0.055
(F9) MPEG2	0.90	0.82	0.19	**0.98**	0.97	0.021
Codec and SI/TI						
(F10) H.264	**0.90**	0.81	0.21	0.95	0.94	0.049
(F11) MPEG2	0.90	0.85	0.13	**0.98**	0.97	0.014

TABLE III
LCC CROSS-VALIDATION RESULTS

	LIVE			Lisbon		
	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ
Only IQA features						
(F1) Mixed	0.51	0.45	0.28	0.82	0.74	0.18
(F2) H.264	0.57	0.45	0.43	0.78	0.72	0.22
(F3) MPEG2	0.54	0.55	0.28	0.85	0.80	0.17
IQA with separation						
(F4) H.264	0.58	0.47	0.40	0.84	0.76	0.20
(F5) MPEG2	0.78	0.70	0.22	0.82	0.74	0.20
IQA + codec features						
(F6) H.264	0.83	0.77	0.23	**0.96**	0.94	0.046
(F7) MPEG2	**0.91**	0.84	0.18	0.96	0.96	0.021
Only codec features						
(F8) H.264	0.86	0.79	0.22	0.95	0.94	0.045
(F9) MPEG2	0.89	0.82	0.19	0.96	0.95	0.026
Codec and SI/TI						
(F10) H.264	**0.91**	0.81	0.20	0.95	0.94	0.045
(F11) MPEG2	0.88	0.84	0.13	**0.97**	0.96	0.019

H.264/AVC and (F5) MPEG-2, separately. (F6) Combining the IQA features with H.264/AVC analysis features (in this case we only extract IQA features from temporally predicted frames). (F7) IQA features with MPEG-2 analysis features (in this case we extract the IQA features only on I-frames). Only using the (F8) H.264/AVC codec analysis features or (F9) MPEG-2 codec analysis features. Lastly, including the simple SI and TI measures with the (F10) H.264/AVC codec analysis features or (F11) MPEG-2 codec analysis features. The idea behind only using IQA features from temporally predicted frames when used in combination with codec features (F6), is that our codec features should be informative about the quality in the I-frames, and the IQA features can then support this with information about the quality in temporally predicted frames.

In all test cases, we used all videos encoded with MPEG-2 and/or H.264/AVC by leaving out 2 contents out of the total 8-12 contents for testing in each cross-validation fold. The remaining videos encoded with MPEG-2 and/or H.264/AVC were used as the training set for that fold of the cross-validation. The experiments were done for all possible content-independent splits between training and test data. This test procedures results in $\binom{n}{k}$ splits, where $k = 2$ and n is the total number of videos. When using content-independent splits *all* videos that have been coded using the same original video is either in the training or in the test set of a cross-validation fold and never split in any way between the two sets.

A. Performance of Proposed Method

The median $\tilde{x}$, mean μ, and standard deviation σ of the Spearman Rank Order Correlation Coefficients (SROCC), of the Linear Correlation Coefficients (LCC), and of the Root Mean Square Error (RMSE) in the different test cases are given in Tables II-IV, respectively. The best $\tilde{x}$ performance for H.264/AVC and MPEG-2 are highlighted in bold.

Based on the results in Tables II-IV, it can be concluded that being able to distinguish between the codecs increases the prediction accuracy (F1-F3 versus F4-F5) e.g. the median SROCC of 0.60 (F3) for the mixed training versus 0.80 (F5) for MPEG-2 in LIVE. For further analysis we shall also apply a statistical analysis of the SROCCs using a multi-comparison analysis of variance (ANOVA) test with a confidence level of 95%. ANOVA testing was first applied to the results using different feature sets for H.264/AVC videos (F2, F4, F6, F8, F10) on both databases. The statistical test showed that the performance of the methods with the feature sets including codec parameters (F6, F8, F10) were statistically significantly superior to the methods with feature sets without codec information (F2, F4).

In the cases where codec analysis features are used (F6-F11), the performance is very good as shown by e.g. the median SROCC of 0.90 (F10) for H.264/AVC in LIVE. Even without any other features than codec analysis features (F8-F9) a median SROCC of 0.88 for H.264/AVC in LIVE was achieved. Adding SI/TI information (F10-F11) slightly increases the robustness of the method as indicated by the low σ values for the Lisbon database and relative low values on the LIVE database. The LCC and RMSE results support the SROCC results, since they generally have the same tendencies for the different feature sets and since the best performance is achieved for the same feature sets (except for F7 and F9).

Regarding the computational complexity: For codec features, both with and without IQA features, we do not use any temporal analysis besides pooling in our method, since our analysis is performed frame by frame. Using SI/TI information is simpler to calculate than most IQA methods since the temporal aspect is only utilized in TI, which is solely based on consecutive frame differences. The most computational complex aspect of our method is the intra-prediction which is comparable in complexity to an H.264/AVC intra only encoder.

B. Comparison to NR Methods

We have compared our method with Video-BLIINDS [25], a recently developed NR PB VQA with a reported median $\tilde{x}$ SROCC of 0.87 on the MPEG-2 videos and 0.84 on H.264/AVC videos in the LIVE database [20]. The testing scheme for the Video-BLIINDS was done with the same cross-validation approach as used in this paper. Compared to our SROCC results, all three versions using codec features (F6-F11) have higher values of $\tilde{x}$. Combining codec features and

TABLE IV
RMSE CROSS-VALIDATION RESULTS

	LIVE			Lisbon		
	$\tilde{x}$	μ	σ	$\tilde{x}$	μ	σ
Only IQA features						
(F1) Mixed	9.8	11	3.2	24	27	9.2
(F2) H.264	10	11	3.5	26	28	9.7
(F3) MPEG2	9.8	11	3.3	22	25	9.3
IQA with separation						
(F4) H.264	15	16	7.5	24	25	10
(F5) MPEG2	11	13	6.5	27	31	11
IQA + codec features						
(F6) H.264	9.2	9.7	1.5	**12**	12	3.9
(F7) MPEG2	**6.6**	7.6	3.4	20	20	2.9
Only codec features						
(F8) H.264	9.8	9.3	3.6	13	13	2.9
(F9) MPEG2	9.3	9.3	1.6	17	17	3.1
Codec and SI/TI						
(F10) H.264	**7.5**	8.1	2.7	13	13	2.5
(F11) MPEG2	7.5	7.3	2.2	**16**	16	2.5

SI/TI (F10-F11) gave a median SROCC of 0.90 for both MPEG-2 and H.264/AVC (Table II). Due to the temporal aspect of Video-BLIINDS that involves motion estimation, it is more complex than our approach even when the TI information is used, which only uses frame differences. On the other hand, it should be noted that Video-BLIINDS is not distortion specific as such, but for the results above it was trained on distortion and codec specific material.

Video-BLIINDS is not capable of distinguishing between the codecs used in the video encoding, so for reference we also report the performance of Video-BLIINDS when testing the general model, published at [33], on the MPEG-2 videos and H.264/AVC videos in the LIVE database with the same splits. This general model has been trained on all videos in the LIVE database, which means that the test videos for each split in the LIVE database have also been used for training. Thus, there is not a clear separation of training and testing, which should expectedly increase the prediction accuracy. The median SROCC, LCC and RMSE for the H.264/AVC and MPEG-2 videos mixed are in this case 0.57, 0.59, and 19, respectively. The median SROCC, LCC and RMSE for only H.264/AVC videos are 0.63, 0.63 and 25. The median SROCC, LCC and RMSE for only MPEG-2 videos are 0.71, 0.70 and 11. Since this performance is much lower than the performance of the model trained on either H.264/AVC or MPEG-2 videos, it clearly indicates, that the prediction accuracy of Video-BLIINDS is higher when the video encoding standard is known or can be distinguished. In [25] the performance of using a state of the art NR IQA as a NR VQA is also reported, yielding a median SROCC value for H.264/AVC of 0.52, which can be compared to the performance when only using IQA features on H.264/AVC videos (F4) with our method, where the median SROCC is equal to 0.67, see Table II.

Also for comparison, a NR PB VQA method with temporal pooling for H.264/AVC videos was presented in [24] using natural scene statistics and Neural Networks. A SROCC of 0.94 was reported using a leave-one-out cross-validation scheme (where 9 contents were used for training *and* validation in each split) on the H.264/AVC subset of the LIVE database. Lastly, in [7] a NR VQA *with* bitstream access is presented where the motion compensated frames are also used and a SROCC of 0.95 for the H.264/AVC videos in the Lisbon database is reported for a single test of equal division between training and test data. When using the codec features (F6, F8, F10), we get similar performance without access to the bitstream.

TABLE V
MEDIAN SROCC FOR FR AND RR METRICS

	PSNR	SSIM	MS-SSIM	VQM	STMAD
LIVE					
Mixed	0.57	0.64	0.88	0.76	**0.93**
H.264	0.71	0.81	0.91	0.86	**0.95**
MPEG-2	0.62	0.79	0.86	0.83	**0.92**
Lisbon					
Mixed	0.89	**0.91**	0.89	0.87	0.87
H.264	0.93	**0.95**	**0.95**	**0.95**	**0.95**
MPEG-2	0.86	0.91	0.91	**0.93**	0.91

C. Comparison to FR and RR Methods

We also tested the performance of well known FR and RR metrics with the same cross-validation procedure. The median of the SROCC and LCC results are given in Tables V and VI, respectively. The metrics include PSNR, SSIM [34], MS-SSIM [34], VQM [35], and STMAD [36]. All of these metrics are FR, except VQM which was used with RR calibration and only uses a set of features extracted from the distorted and original videos to predict the video quality. It should be noted, that for the LCC values in Table VI, a nonlinear fit between the predicted and actual values was performed before the calculation of the LCC. (For the LCC values in Table III, a non-linear fit was not performed but our use of SVR implies a non-linear mapping.) In Tables V and VI, STMAD seems to perform the best of the FR and RR metrics on the LIVE database, while VQM has the overall best performance measured on the Lisbon database.

When only using IQA features (F1-F3 in Tables II-III), we achieve similar performance as the PSNR measure wrt. median SROCC. When using codec features (F6-F11) we perform better than PSNR, SSIM, and MSSIM wrt. median SROCC. Comparing with VQM and STMAD, which use temporal features, we perform better wrt. median SROCC in all cases except for STMAD on the LIVE videos (Table V).

Using multi-comparison ANOVA with a confidence level of 95% on the SROCC performance of our proposed method with codec information on H.264/AVC videos (F6, F8, F10) versus the FR and RR methods in the LIVE database revealed that STMAD was superior i.e. better with statistical significance, to all other methods except the proposed method with codec and SI/TI information (F10) and MS-SSIM. Also, both MS-SSIM and the proposed method with codec and SI/TI information (F10) were superior to the performance of PSNR using the same statistical test. Multi-comparison ANOVA on the SROCC values for Lisbon showed that all methods except SSIM had superior performance compared to PSNR.

Using the same ANOVA test for the performance on MPEG-2 videos in the LIVE database revealed that the proposed method for F7 and F11 and STMAD were the only methods superior to both PSNR and SSIM, while the proposed method

TABLE VI
MEDIAN LCC FOR FR AND RR METRICS

	PSNR	SSIM	MS-SSIM	VQM	STMAD
LIVE					
Mixed	0.64	0.76	0.82	0.74	**0.95**
H.264	0.72	0.82	0.87	0.84	**0.94**
MPEG-2	0.64	0.76	0.84	**0.92**	0.91
Lisbon					
Mixed	0.84	**0.90**	0.84	0.86	0.83
H.264	0.88	0.94	0.95	**0.96**	0.93
MPEG-2	0.82	0.87	0.86	**0.94**	0.89

with F9, MS-SSIM and VQM were only superior to PSNR. For the MPEG-2 videos in the Lisbon database the statistical test showed that the proposed methods using codec information features (F7, F9, F11) were superior to all the FR/RR metrics reported in Table V. To sum up comparisons of our method using codec identification and Codec features and SI/TI (F10, F11) versus the best FR method STMAD; for MPEG-2 on Lisbon (F11) is superior to STMAD, while for the other ANOVA tests there were no significant difference.

D. Robustness

Our results show the codec dependency of the proposed and state of the art methods. Another type of dependency is content dependency. To some extent the standard deviations of our results reflect this aspect, since 2 contents have been used only in the test set of each fold of the cross-validation. Thus, relatively low standard deviations suggest that a method is robust and not content dependent. As can be seen from the results, the combination of codec features and SI/TI features generate high median values between 0.90 to 0.98 and with low standard deviations on Lisbon (between 0.014 to 0.049) while relatively low, but slightly higher on LIVE (between 0.13 and 0.21). The higher values for LIVE indicates that there is still room for improvement.

E. Cross-database Performance

To further validate our method and the independence of a particular test database, we trained our algorithm on all 40 H.264/AVC videos in the LIVE database and tested it on all 56 H.264/AVC videos in the Lisbon database and vice versa. Since there is no overlap in videos between the two databases, this split is also content-independent. Even though the two databases are very different from each other (different encoder settings, video resolutions and scoring methodology) we achieved promising results with a SROCC equal to 0.79, a LCC equal to 0.75, and a RMSE equal to 28, when training on the LIVE database and testing on the Lisbon database.

This can be compared to the general model of the Video-BLIINDS [33], where the quality predictions of the H.264/AVC videos in the Lisbon database result in a SROCC equal to 0.52, a LCC equal to 0.52 and a RMSE equal to 41. When we train our model on the Lisbon database and test on the LIVE database the quality predictions result in a SROCC equal to 0.64, a LCC equal to 0.61, and a RMSE equal to 20.

F. Considering HEVC videos

We briefly consider the scenario where HEVC encoding may be used in addition to the MPEG-2 and H.264/AVC encoding. For initial testing purposes, we encoded all the original videos from the LIVE database with HEVC encoding at four different bitrates. First, we consider MPEG-2 videos and HEVC videos using the same method for separating MPEG-2 and H.264/AVC videos as presented in [10] and as briefly described in Section II, again with 100% accuracy.

Next, we considered distinguishing between H.264/AVC videos and HEVC videos. This is more difficult due to the similarities of the two codecs. Even so, if the H.264/AVC analysis is performed on HEVC videos, the confidence vector $\mathbf{c}$ (16) will have lower differences between elements corresponding to I-frames and elements corresponding to inter-predicted frames. In our test described above, we observed that applying the GOP size estimation to H.264/AVC videos will in most cases produce high GOP length estimates for the HEVC videos.

In this small experiment, 75% of the HEVC encoded videos had an estimated GOP of 40 or above when using the H.264/AVC analysis method without an offset, and we use this to identify them as HEVC videos (as opposed to MPEG2 and H.264/AVC). The GOP size for the rest of the HEVC encoded videos in the experiment were estimated to 16, which was the actual GOP size used in the encoding. For the latter videos, we also estimated the QP parameters using the H.264/AVC analysis. The Root Mean Squared Error (RMSE) of the estimated QP values of those HEVC I-frames was 2.88, which could indicate that our H.264/AVC analysis does provide meaningful information for these HEVC encoded sequences. This is due to the fact, that the HEVC encoding resembles H.264/AVC encoding in many aspects. To be able to also handle HEVC encoded videos in a more robust manner, further work would be needed, but this is left for further study.

VII. CONCLUSION

In this paper, the architecture of a Pixel-Based NR VQA method for H.264/AVC and MPEG-2 videos based on codec analysis was presented. To achieve this, we presented methods to estimate objective measures such as quantization and PSNR. Contrary to other state of the art approaches using codec information, the proposed method is only based on the decoded video and it does not require access to the video bitstream. It was also demonstrated how the proposed method can be used to produce codec information for H.264/AVC encoded videos.

The features based on the video codec analysis were mapped to a quality score using SVR. Testing was performed on two video quality databases and it was shown that the proposed PB codec analysis is robust, even when intra-prediction and deblocking is enabled in H.264/AVC videos. The results show statistically significant improvement when distinguishing between codecs in NR VQA. Combining codec and SI/TI features in the NR VQA achieved median SROCC values of 0.90 for the LIVE database and 0.95 - 0.98 for the other (Lisbon) database. Furthermore, the results show that the proposed method is performing well when compared to state of the art NR, RR, and FR VQA methods.

REFERENCES

[1] J. G. Apostolopoulos and A. R. Reibman, "The challenge of estimating video quality in video communication applications," *IEEE Signal Process. Mag.*, pp. 156–160, May 2012.

[2] G. Valenzise, S. Magni, and M. Tagliasacchi, "No-reference pixel video quality monitoring of channel-induced distortion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 605–618, 2012.

[3] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. C. Cosman, and A. R. Reibman, "A versatile model for packet loss visibility and its application to packet prioritization," *IEEE Trans. Image Process.*, vol. 19, pp. 722–735, 2010.

[4] F. Yang, S. Wan, Q. Xie, and H. R. Wu, "No-reference quality assessment for networked video via primary analysis of bit stream," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, pp. 1544–1554, 2010.

[5] A. Ischigaya, Y. Nishida, and E. Nakasu, "Nonreference method for estimating PSNR of MPEG-2 coded video by using DCT coefficients and picture energy," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 817–826, 2008.

[6] A. Eden, "No-reference estimation of the coding PSNR for H.264-coded sequences," *IEEE Trans. Consum. Electron.*, vol. 53, no. 2, pp. 667–674, 2007.

[7] T. Brandão and M. P. Queluz, "No-reference quality assessment of H.264/AVC encoded video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1437–1447, 2010.

[8] T. Na and M. Kim, "A novel no-reference PSNR estimation method with regard to de-blocking filtering effect in H.264/AVC bitstreams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 320–330, 2014.

[9] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Estimating QP and motion vectors in H.264/AVC video from decoded pixels," in *Proc. 2nd ACM workshop Multimedia forensics, security, intelligence*, New York, 2010, pp. 89–92.

[10] S. Forchhammer, H. Li, and J. D. Andersen, "No-reference analysis of decoded MPEG images for PSNR estimation and post-processing," *Journal Visual Comm. Image Representation*, vol. 22, no. 4, pp. 313–324, 2011.

[11] J. Søgaard, S. Forchhammer, and J. Korhonen, "No-reference video quality assessment using MPEG analysis," in *Proc. Picture Coding Symposium*, San Jose, 2013, pp. 161 – 164.

[12] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, 2012.

[13] *Recommendation ITU-T P.910: Subjective video quality assessment methods for multimedia applications*, Int'l Telecom. Union Std., 2008.

[14] S. R. Gunn, "Support vector machines for classification and regression," University of Southampton, School of Electronics and Computer Science, ISIS technical report, May 1998.

[15] S. Tubaro, M. Tagliasacchi, A. Allam, P. Bestagini, and S. Milani, "Video codec identification," in *ICASSP, IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, 2012.

[16] I. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. Wiley, 2010.

[17] R. Reininger and J. D. Gibson, "Distributions of the two-dimensional DCT coefficients for images," *IEEE Trans. Commun.*, vol. 31, no. 6, pp. 835–839, 1983.

[18] Y. Altunbasak and N. Kamaci, "An analysis of the DCT coefficient distribution with the H.264 video coder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 3, Montreal, 2004, pp. 177–180.

[19] N. Kamaci, Y. Altunbasak, and R. M. Mersereau, "Frame bit allocation for the H. 264/AVC video coder via Cauchy-density-based rate and distortion models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 994–1006, 2005.

[20] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Trans. Image Process.*, vol. 19, pp. 1427–1441, 2010.

[21] P. Gastaldo, S. Rovetta, and R. Zunino, "Objective assessment of MPEG-video quality: a neural-network approach," in *Proc. Int'l. Joint Conf. Neural Networks*, vol. 2, Washington, DC, 2001, pp. 1432–1437.

[22] S. Mohamed and G. Rubino, "A study of real-time packet video quality using random neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1071–1083, 2002.

[23] P. Le Callet, C. Viard-Gaudin, and D. Barba, "A convolutional neural network approach for objective video quality assessment," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1316–1327, 2006.

[24] K. Zhu, K. Hirakawa, V. Asari, and D. Saupe, "A no-reference video quality assessment based on laplacian pyramids," in *Proc. IEEE Int'l Conf. Image Process.*, Melbourne, 2013, pp. 49 – 53.

[25] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind prediction of natural video quality," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1352–1365, 2014.

[26] S. Argyropoulos, A. Raake, M.-N. Garcia, and P. List, "No-reference video quality assessment for SD and HD H.264/AVC sequences based on continuous estimates of packet loss visibility," in *Proc. Third Int'l Workshop Quality Multimedia Experience*, Mechelen, 2011, pp. 31 – 36.

[27] N. Staelens, D. Deschrijver, E. Vladislavleva, B. Vermeulen, T. Dhaene, and P. Demeester, "Constructing a no-reference H.264/AVC bitstream-based video quality metric using genetic programming-based symbolic regression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1322–1333, 2013.

[28] M. Narwaria and W. Lin, "SVD-based quality metric for image and video using machine learning," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 2, pp. 347–364, 2012.

[29] M. Narwaria, W. Lin, and A. Liu, "Low-complexity video quality assessment using temporal quality variations," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 525–535, 2012.

[30] J. Park, K. Seshadrinathan, S. Lee, and A. C. Bovik, "Video quality pooling adaptive to perceptual distortion severity," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 610–620, 2013.

[31] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intelligent Systems Technology*, vol. 2, pp. 27:1–27:27, 2011, http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[32] *H.264/AVC Software Coordination*, Joint Video Team (JVT) Std., 2007. [Online]. Available: http://iphome.hhi.de/suehring/tml/

[33] M. Saad. (2014) Video-bliinds software. [Online]. Available: http://live.ece.utexas.edu/research/quality/VideoBLIINDS_Code_MicheleSaad.zip

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[35] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Trans. Broadcast.*, vol. 50, no. 3, pp. 312–322, 2004.

[36] P. Vu, C. Vu, and D. Chandler, "A spatiotemporal most-apparent-distortion model for video quality assessment," in *Proc. IEEE Int'l Conf. Image Processing*, Brussels, 2011, pp. 2505–2508.

Jacob Søgaard received the B.S. degree in engineering, in 2010, and the M.S. degree in engineering, in 2012, from the Technical University of Denmark, Lyngby, where he is currently pursuing his Ph.D. degree with the Coding and Visual Communication group at the Department of Photonics.

His research interests include image and video coding, image and video quality assessment, visual communication, and machine learning for Quality of Experience purposes.

Søren Forchhammer (M'04) received the M.S. degree in engineering and the Ph.D. degree from the Technical University of Denmark, Lyngby, in 1984 and 1988, respectively. Currently, he is a Professor with DTU Fotonik, Technical University of Denmark, where he has been since 1988. He is Head of the Coding and Visual Communication Group at DTU Fotonik. His main interests include source coding, image and video coding, video quality, distributed video coding, processing for image displays, and visual communications.

Jari Korhonen (M'05) received his M.Sc. (Eng.) degree in information engineering from University of Oulu, Finland, in 2001, and Ph.D. degree in telecommunications from Tampere University of Technology, Finland, in 2006. He is currently an Assistant Professor at DTU Fotonik, Technical University of Denmark, since 2010. His research interests cover both telecommunications and signal processing aspects in multimedia communications, including visual quality assessment.