A Regression-Based Framework for Estimating the Objective Quality of HEVC Coding Units and Video Frames

Tamer Shanableh

Department of Computer Science and Engineering

American University of Sharjah, UAE

Fax: +971 6 515-2979

tshanableh@aus.edu

Abstract

A no-reference objective quality estimation framework is proposed. The framework is suitable for any block-based video codec. In the proposed solution, features are extracted from coding units and summarized to form features at frame levels. Stepwise regression is used to select the important feature variables and reduce the dimensionality of feature vectors. Thereafter, a polynomial regression-based approach is used to model the nonlinear relationship between the feature vectors and the true objective quality values. Such values are estimated for coding units and video frames. The proposed framework is implemented using MPEG-2 and HEVC. The objective quality estimation results are compared against an existing state-of-the-art solution and quantified using the Pearson correlation factor and the root mean square error measure.

Index Terms: PSNR estimation; SSIM estimation; machine learning; regression analysis; video codecs; video compression

1. Introduction

In video broadcasting and IPTV, it is desired to monitor the quality of delivered services. There is a need in such applications to automatically monitor and estimate the quality of compressed video due to the distortions caused by lossy coding, transmission errors and potential intermediate video transcoding.

Objective quality estimation of compressed video falls into two main categories; Reduced Reference (RR) and No Reference (NR) estimations. In the RR category, special information is extracted from the original frames and subsequently made available for PSNR estimation. On the other hand, such information is not available for objective quality estimation in the NR category. Therefore such category is less accurate and more challenging.

An example of the RR estimation is the use of distributed source coding techniques where the encoder transmits the Slepian-Wolf syndrome a feature vector representing the original video frame using a LDPC encoder. The receiver reconstructs the side information of the received frame from the Slepian-Wolf bistream. Therefore the original feature vector is not transmitted and therefore the overall bit rate is reduced [1].

On the other hand, no-reference objective quality estimation can be applied to a video frame or to the whole video sequence. In [2] features are extracted from the whole sequence and compared against a dataset of features belonging to sequences of different spatio-temporal activities. Some solutions are applied to predict the PSNR of both video sequences and frames [3]. It was also reported that the PSNR of a video sequence can be estimated based on the average bitrate and mean quantization parameter of the I-frames only [4].

Estimating objective quality at a frame level can make use of the distribution statistics of Discrete Cosine Transformation (DCT) Coefficients. In [5, 6] it was proposed to estimate the quantization error from the statistics of DCT coefficients to estimate the PSNR. It is noted that DCT coefficients follow a Laplacian probability distribution. The Lambda Laplacian distribution parameter is estimated for each DCT frequency band separately.

More recently, a method for estimation the PSNR of H.264/AVC video frames which considers both the deblocking filtering effect and the quantization error is proposed in [7].

Degradations due to transmissions are also taken into account in estimating the video quality. For instance, bit stream information, quantization distortions, packet losses and temporal effects of the human visual system are all used for estimating video quality [8, 9].

In addition to estimating the video quality for broadcasting and streaming applications, quality estimation is useful in other scenarios as well. For instance, the quality of a surveillance video is assessed prior to admitting it to as a legal evidence in a court of law. In this a case, it is preferred to estimate the quality of the video at a sub-frame level or at a macroblock level. This is needed as some regions of a compressed frame might be of higher interest to a jury. MB-level PSNR estimation is reported in [10, 11] and MB-level SSIM estimation is reported in [12].

More recently, a no-reference PSNR estimation method for High Efficiency Video Coding (HEVC) was proposed in [13]. The method estimates the PSNR at a frame-level based on a Laplacian mixture distribution. The solution computes the distribution parameters of residual DCT coefficients in different quadtree depths and different types of coding units (CUs). Since some DCT bands might be all zeros, an exponential regression solution is used that takes into account the CU coding depths. While the prediction results are very accurate, one limitation of such a solution is that it assumes a fixed value of QP. Hence it is not suitable for constant bitrate coding.

In this work, we approach objective quality estimation from a regression-based perspective. We propose a generic framework which is suitable for any block-based video codec. The proposed solution is applied to MPEG-2 and HEVC. The objective quality is estimated at both a CU/MB level and at a frame level. The objective quality metrics used in this work are PSNR and Structural Similarity Indices (SSIM) [r14]. Advantages of the proposed solution are its generic framework and suitability for estimating the PSNR of videos coded with constant bitrates. To the best of the author's knowledge, this solution is the first to estimate the PSNR and SSIM at coding unit (CU) level in HEVC.

This paper is organized as follows. Section 2 briefly introduces HEVC and its new portioning feature which is known as the Coding Units (CUs). Section 3 introduces the proposed solution and objective quality estimation framework which is used to predict PSNR and SSIM values for CUs and frames. Section 4 reviews the tools used in the proposed systems; namely, stepwise regression and polynomial regression. The experimental results are given in Section 5 and Section 6 concludes the paper.

2. HEVC Coding Units

The Joint Collaborative Team on Video Coding (JCT-VC) proposed and developed the High Efficiency Video Coding (HEVC) standard [15]. The objective of which is to offer a substantially higher compression capability in comparison to existing standardized codecs. HEVC also targets

new applications, such as beyond high-definition spatiotemporal resolutions, various samplings formats and color spaces.

One of the main features of HEVC is the frame partitioning which results in higher prediction accuracy. A frame is divided into square blocks known as coding units (CUs). The maximum allowed size is 64 × 64 for the luma component and the minimum size, on the other hand, is 8x8. The syntax of each CU indicates the type of prediction, the transform unit (TU) sizes and the types of the prediction units (PU) used. The syntax also defines if a CU is coded in split mode. The largest CU is said to have the depth of 0 and if it is further split then the four resultant CUs have a depth of 1, and so forth. The partitioning used for motion estimation and compensation is instructed by the size of the PUs. Several PU sizes are allowed as follows; 2N × 2N, 2N×N, N×2N, N×N, 2N×nU, 2N×nD, nL×2N and nR×2N. Further details about HEVC can be found in [16].

In this work, we are particularly interested in CUs and their PUs as features are collected from these coding and partitioning units.

3. Proposed system

As mentioned previously, in this work, we use a polynomial regression-based approach to predict the objective quality of both video coding units (CUs) and full frames.

The use of machine learning techniques for the prediction of objective video quality is reported in the literature for a number of video codecs. For instance, the work in [17] employed artificial neural networks to predict the quality of MPEG-2 videos using features that are extracted directly from the video streams. Likewise, a number of machine learning techniques including SVM and Bayes classifiers are used to classify the quality of coded MPEG-2 macroblocks into various PSNR classes [18]. Additionally, linear regression was used to predict the objective quality of videos coded using the H.264/AVC codec. The features are extracted from the video bitstreams and consist of motion and coding parameters [19, 20]. More recently, machine learning is used to predict several objective quality metrics with reasonable accuracy for the H.264/AVC codec [21]. The work is further enhanced using artificial neural networks with reduced complexity as reported in [22].

The proposed system is divided into two stages, training and testing. In the training phase, video bit streams and their PSNR/SSIM values are used to build a regression model. The training system is illustrated in Figure 1. CU features are extracted from a coded video stream. These features are further summarized on frame basis to compute frame-level features. That is, all the features of all CUs belonging to one frame are summarized by computing their mean and standard deviation. The

numerical summarization is performed in terms of central tendency and dispersion. Each CU and each frame is represented by one feature vector. The dimensionality of the CU and frame feature vectors are reduced using a statistical procedure known as stepwise regression [23] which is explained in the next section. Stepwise regression retains the features that affect the response variable the most, where the response variable in this case is the true PSNR/SSIM value. Hence, the inputs to this procedure are both the feature vectors and their corresponding PSNR/SSIM values. Clearly, the PSNR/SSIM values are not available during the testing phase, hence, in the training phase, the indices of the feature variables selected by the stepwise procedure are stored and reused in the testing phase. Once reduced in dimensionality, the feature vectors and their corresponding PSNR/SSIM values are used to compute a regression model using polynomial regression [24], which is reviewed in Section 4. The model weights are stored and used in the testing phase as well.

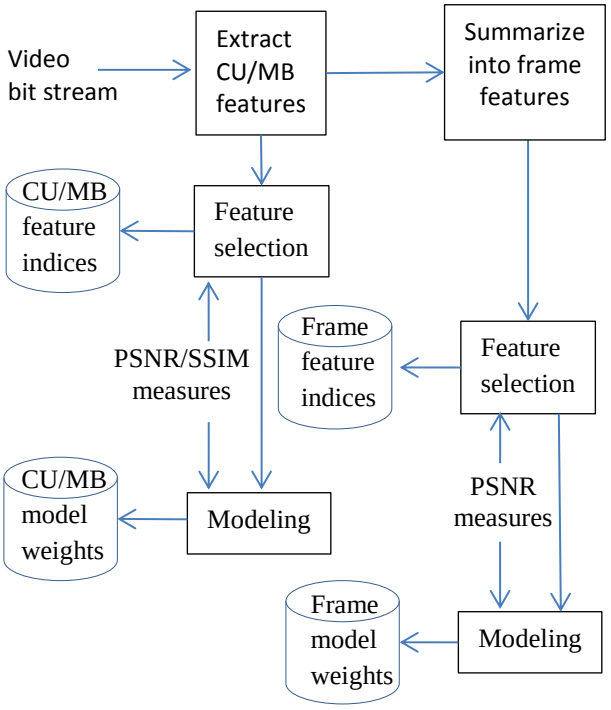

Figure 1. Proposed training framework for objective quality estimation.

In the testing phase, CU features are extracted from a coded video bit stream and summarized into frame-level features. This arrangement is exactly the same is that used in the training phase. The proposed testing system is illustrated in Figure 2. The dimensionality of the feature vectors is reduced simply by retaining the variables corresponding to the stored indices from the training stage.

A CU and/or a frame PSNR/SSIM is predicted using the stored polynomial regression model. Again, the algorithm used for modeling and prediction is explained in Section 4.

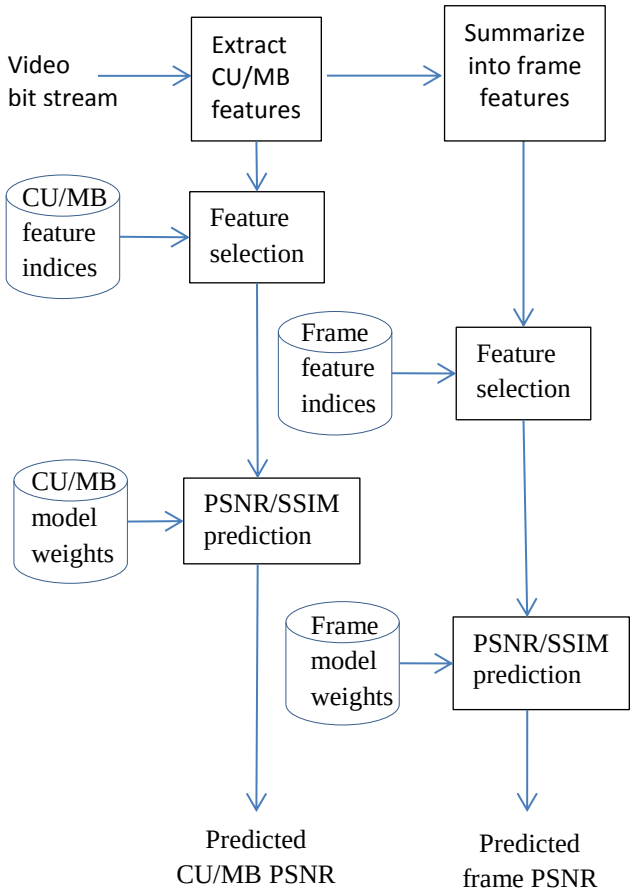

Figure 2. Proposed testing framework for objective quality estimation.

It is worth to mention that the proposed system is not restricted to the objective quality prediction of HEVC videos. It is generic enough to work with any block-based video codec. For instance in this work we also consider the prediction of frame-level PSNR values for MPEG-2 videos. In a similar arrangement to the one discussed above, the MB-level features are numerically summarized into frame features and used to predict PSNR values. Such an approach builds on top of the work previously reported by the author for the prediction of MB-level PSNR values for MPEG-2 video [10].

The HEVC CU features that are extracted from a video bit stream are listed in Table 1. Clearly, all of these features can be extracted in both the training and the testing phases as access to the original

raw video is not required. This process is carried out at the decoder, hence, there is only access to quantized coefficients and syntax elements.

ID		Feature description
1	CU features	Total number of bits in a CU
2		X coordinate of a CU in pixels
3		Y coordinate of a CU in pixels
4		Quantization step size
5		Quantization step size of previous CU
6		Variance of DCT coefficients
7		Mean of DCT coefficients
8		Skewness of DCT coefficients
9		Variance of residual pixels
10		Mean of residual pixels
11		Skewness of DCT residual pixels
12		Variance of reconstructed pixels
13		Mean of reconstructed pixels
14		Skewness of reconstructed pixels
15		Variance of prediction source
16		Mean of prediction source
17		Skewness of prediction source
18		Total number of partitions in a CU
19	PU features	Coding depth
20		Partition type (2Nx2N, 2NxN,…)
21		Partition width
22		Partition height
23		Coding mode
24		Transformation index
25		Merge Flag
26		Merge Index
27		Inter prediction direction
28		Coded block flag
29		xMV of list 0
30		yMV of list 0
31		Difference xMV of list 0
32		Difference yMV of list 0
33		xMV of list 1
34		yMV of list 1
35		Difference xMV of list 1
36		Difference yMV of list 1
37		Variance of residual pixels
38		Mean of residual pixels
39		Skewness of DCT residual pixels
40		Variance of reconstructed pixels
41		Mean of reconstructed pixels
42		Skewness of reconstructed pixels
43		Variance of prediction source
44		Mean of prediction source
45		Skewness of prediction source

Table 1. HEVC CU feature variables extracted from video bit streams.

Since each CU can be partitioned into many PUs, further features are extracted for each PU. Eventually the values of such variables are merged by computing their means and variances. In this work, the operations of the encoder are not modified. The HEVC encoder selects the best partitioning for a CU based on rate-distortion optimization. The features are extracted from the CU syntax elements and the individual PUs. Then, the features of the PUs are statistically summarized. The CU features and the summarized PUs features are concatenated into one feature vector. Thus, the features at index 19 to 45 in Table 1 are repeated twice, once as mean values and once as variances. This brings the total number of features to 72. For intra-coded slices or CUs, the motion information is set to 0 and the coding type is signaled in the feature vector.

As mentioned previously, all the features of all CUs belonging to one frame are summarized by computing their mean and standard deviation. This results in one feature vector which is used for frame-level quality prediction. The next section will present a solution in which the dimensionality of the feature vectors is reduced.

As for the MB-level MPEG-2 feature variables, we use the ones reported in [10]. Basically they are similar in concept to the variables listed in Table 1. More specifically they are based on syntax elements, motion information, coding types and texture statistical measures.

4. System modeling

This section reviews the tools used in the proposed solution and formulates the problem using polynomial regression.

4.1 Selecting the important features

Important features can be objectively selected using the stepwise regression procedure. In this work we use stepwise regression to select important features and to reduce the dimensionality of the feature vectors. As mentioned previously, the stepwise regression procedure is applied during the training phase only. The indices of the selected feature variables are then stored and used in the testing phase.

We treat the feature variables $f_1, f_2, \ldots, f_m$ as predictors where the subscript refers to the feature ID as outlines in Table 1 and m is the total number of features in each feature vector. All of the feature vectors belonging to the training set are organized into a features matrix. Likewise, the actual PSNR/SSIM values are treated as a response variable, p.

The stepwise regression procedure is described in [23] using the following procedure. In the first step, the procedure tests all possible one-predictor regression models in an attempt to find the predictor that has the highest correlation with the response variable. The model is of the form:

$$\hat{p} = \beta_0 + \beta_1 f_i \qquad (1)$$

A hypothesis test is conducted for each model where $H_0: \beta_1 = 0$ and $H_1: \beta_1 \neq 0$. The test is conducted using the well-known *T* test at a specific level of significance, for example $\alpha = 0.1$. The predictor that generates the largest absolute *T* value is selected. Refer to this predictor as f_1.

In the second step, the remaining m-1 predictors are scanned for the best two-predictor regression model of the form:

$$\hat{p} = \beta_0 + \beta_1 f_1 + \beta_2 f_i \qquad (2)$$

This is achieved by testing all two-predictor models containing f which was selected from the first step. The *T* value of the m-1 models are computed for $H_0: \beta_2 = 0$. The predictor that generates the highest absolute *T* value is retained, refer to this predictor as f_2.

Now that $\beta_2 f_2$ is added to the model, the procedure goes back and reexamines the suitability of including β_1 in the model. If the corresponding *T* value becomes insignificant (i.e. the alternative hypothesis H_1 is rejected.), f_1 is removed and the predictors are searched for a variable that generates the highest T value in the presence of $\beta_2 f_2$. In the third step, remaining m-2 predictors are scanned for the best three-predictor regression model of the form:

$$\hat{p} = \beta_0 + \beta_1 f_{1+} \beta_2 f_{2+} \beta_3 f_i \qquad (3)$$

At this point, the *T* values for f_1 and f_2 are computed and if any of them became insignificant after adding f_3 then the corresponding variable is removed from the model. The procedure repeats the above steps until no additional predictors are added or removed from the model.

Again, this procedure is applied to the feature vectors prior to system modeling which is explained next. Hence only the important features are retained and used for modeling. The number of retained features is fixed for all feature vectors representing a CU or a video frame.

In the experimental results section, we show the results of applying the stepwise regression procedure to the feature vectors of the CUs and the video frames.

4.2 Polynomial regression

In this work, PSNR/SSIM prediction is formulated using polynomial regression [13]. In the formulation, the extracted feature vectors are referred to as (**F**) and the actual PSNR/SSIM values are referred to as (**p**).

As mentioned in the previous section, the dimensionality of the feature vectors is reduced using stepwise regression. According to Figure 1, if the PSNR/SSIM is predicted at a CU-level then these feature vectors belong to individual CUs. Otherwise, if the PSNR/SSIM is predicted at frame-level then the feature vectors belong to individual frames. These feature vectors or predictors of n CUs/frames are arranges into one feature matrix after applying the stepwise regression procedure. This matrix is denoted by **F** as shown in Equation (4)

$$\mathbf{F} = \begin{bmatrix} \mathbf{f_{1,1}} & \mathbf{f_{2,1}} \cdots & \mathbf{f_{m,1}} \\ \vdots & \vdots & \vdots \\ \mathbf{f_{1,n}} & \mathbf{f_{2,n}} \cdots & \mathbf{f_{m,n}} \end{bmatrix} \tag{4}$$

The subscripts of the matrix elements $f_{j,i}(j = 1..m, i = 1..n)$ indicate the index of feature variables and the corresponding CU/ frame index respectively.

In polynomial regression, a nonlinear mapping is performed between the feature matrix **F** and the response variable **p**. As such, the dimensionality of the rows in matrix **F** is expanded into an r^{th} order. This is achieved using a reduced model polynomial expansion [24]. The expanded feature matrix is referred to as $\mathbf{X} \in \mathcal{R}^{nxk}$ where k is the dimensionality of the expanded feature vectors. According to [24], the dimensionality of the expanded feature vector is defined by $k = 1 + r + m(2r - 1)$. Where m denotes the number of features variables. In this work we use a second order expansion. A second order expanded feature vector consists of the following terms:

$$\text{expand}(\mathbf{f}) = [1,\ f_j^2 \mid 1 \le j \le m\,, \\ (f_1 + f_2 + \cdots + f_m)^2, f_j(f_1 + f_2 + \cdots + f_m) \mid 1 \le j \le m\,] \tag{5}$$

For example, if a feature vector contains two variables f_1 and f_2 then the expanded feature vector will contain the following terms: $[1, f_1, f_2, f_1 f_2, f_1^2, f_2^2, (f_1 + f_2), (f_1 + f_2)^2, f_1(f_1 f_2), f_2(f_1 f_2)]$.

A least-squared error objective criterion is used to perform the mapping between **X** and **p** as follows:

$$\boldsymbol{\alpha}^{\text{opt}} = \arg_{\boldsymbol{\alpha}} \min \|\mathbf{X}\boldsymbol{\alpha} - \mathbf{p}\|_2 \tag{6}$$

Where $\|.\|_2$ denotes the l_2 norm.

To predict a PSNR value, the feature vectors are extracted from the video bit stream. The feature vectors are then arranged into a feature matrix and expanded to the second order. This results in the **X** matrix. The feature matrix is multiplied by the model weights $\boldsymbol{\alpha}^{\text{opt}}$ to compute the predicted PSNR values $\hat{\mathbf{p}}$ as follows:

$$\hat{\mathbf{p}} = \mathbf{X} * \boldsymbol{\alpha}^{opt} \tag{7}$$

5. Experimental results:

The proposed system is implemented using HEVC reference software HM13 [25] and MPEG-2 [26]. The used video sequences, their resolutions and bitrates are listed in Table 2.

ID	Name	Resolution	Bitrate-1 (Mb/s)	Bitrate-2 (Mb/s)
1	BasketBallPass	416x240 (60Hz)	1	3
2	BQSquare	416x240 (60Hz)	1	3
3	BasketBallDrill	832x480 (30Hz)	1.5	3.5
4	Horses	832x480 (30Hz)	1.5	3.5
5	Mall	832x480 (30Hz)	1.5	3.5
6	Party	832x480 (30Hz)	1.5	3.5
7	City	1280x720 (60Hz)	2.0	6
8	BasketBallDrive	1920x1080 (50Hz)	3.0	6
9	Cactus	1920x1080 (50Hz)	3.0	6
10	Crew	352x288 (25Hz)	0.5	1.5
11	Coastguard	352x288 (25Hz)	0.5	1.5
12	Foreman	352x288 (25Hz)	0.5	1.5
13	Football	352x288 (25Hz)	0.5	1.5
14	Walk	352x288 (25Hz)	0.5	1.5

Table 2. Test video sequences and their properties

In the experiments to follow, we report the results using the bitrates of the first set, bitrate-1. We then repeat the results using bitrate-2 to confirm that the proposed solution works for both sets of bitrates.

In HEVC, the coding structure used is IPPPPP… with 4 reference frames. The maximum CU size is set to 64x64. The asymmetric motion partitions tool and the adaptive loop filter tool are both enabled. The default fast motion estimation (a modified EPZS) and fast mode decisions are used. Constant bitrate coding at CU level is enabled. The coding configuration for MPEG-2 is similar. However, only one reference frame is used and the MB size is 16x16. The same constant bitrate values are used in both coders.

We start by showing the results of applying the stepwise regression procedure for the prediction of frame-level PSNR values. Table 3 lists the feature variables that are retained for the video sequences listed in Table 2. The videos are compressed using HEVC using the above-mentioned coder configuration.

Sequence ID														Feature	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	ID	
		x				x			x			x	x	1	CU features
														2	
														3	
	x	x		x	x	x	x	x	x	x		x	x	4	
x	x		x	x	x		x	x			x			5	
					x							x		6	
														7	
					x			x						8	
												x		9	
x														10	
														11	
x		x	x	x			x							12	
x			x	x	x									13	
x														14	
														15	
														16	
		x		x							x			17	
														18	
	x				x				x				x	19	Mean of PU features
						x								20	
			x	x							x			21	
														22	
														23	
x			x		x	x	x							24	
		x			x	x	x				x	x		25	
	x				x									26	
										x				27	
		x	x				x	x						28	
x	x	x	x	x		x	x							29	
			x			x			x	x	x			30	
												x		31	
									x					32	
										x			x	33	
														34	
														35	
														36	
														37	
														38	
														39	
			x											40	
x					x		x	x						41	
x					x			x	x		x			42	
														43	
				x										44	
						x	x							45	
							x							46	Variance of PU features
							x				x			47	
							x			x			x	48	
						x								49	
	x		x											50	
				x				x				x		51	
														52	
x				x	x		x	x		x		x		53	
														54	
													x	55	
						x			x					56	
		x						x	x			x		57	

			x		x									58	
														59	
														60	
														61	
														62	
														63	
														64	
														65	
														66	
														67	
					x									68	
		x												69	
									x					70	
x			x				x							71	
					x									72	
11	6	9	12	10	15	10	14	9	8	7	7	9	6	Total no.	

Table 3. Frame-level feature selection using stepwise regression.

As shown in the table, the number of features retained by the stepwise regression procedure ranges from 6 to 15. The retained features are used for building the regression model as explained in Section 4 above. When repeating this experiment at the CU level, as opposed to the frame level, it was noticed that the number of retained features are higher as shown in Table 4.

Sequence ID	Number of features
1	24
2	31
3	35
4	33
5	33
6	35
7	30
8	30
9	39
10	23
11	24
12	23
13	13
14	23

Table 4. CU-level feature selection using stepwise regression.

The increase in number of features is understood, as at the frame level, all of the CU data is summarized. As such the resultant data is rich and representative. Whereas at the CU level, the features available are not as representative.

The proposed system is assessed and compared against existing work through the use of two performance attributes; prediction accuracy and prediction consistency. These attributes are proposed by the Video Quality Experts Group (VQEG) [27]. The Pearson linear correlation coefficient is used to assess the prediction accuracy and the Root Mean Square Error (RMSE)

measure is used to assess the prediction consistency. Cleary, the objective is to minimize the RMSE and to maximum the correlation.

The frame-level PSNR prediction results using the proposed system are reported in Table 5. The feature vectors of each video sequence were divided into 25% for training and 75% for testing. Table 5, shows the prediction results using MPEG-2 and HEVC coded sequences.

	MPEG2		HEVC	
Seq. ID	RMSE (dB)	Corr.	RMSE (dB)	Corr.
1	0.7	0.99	0.6	0.98
2	0.1	0.98	0.2	0.99
3	0.2	0.93	0.28	0.94
4	0.2	0.99	1.1	0.92
5	0.1	0.98	0.3	0.97
6	0.2	0.98	0.6	0.96
7	0.12	0.96	0.5	0.9
8	0.1	0.99	0.5	0.95
9	0.1	0.97	0.4	0.96
10	0.24	0.98	0.5	0.969
11	0.12	0.981	0.93	0.954
12	0.23	0.938	0.46	0.956
13	0.5	0.979	0.86	0.98
14	0.35	0.975	0.6	0.96
Avg.	0.23	0.97	0.56	0.96

Table 5. Frame-level PSNR prediction results using the proposed system.

The prediction results indicate that the proposed system is capable of predicting the frame-level PSNR values with high accuracy. The results also indicate that the proposed system is generic enough to incorporate different block-based video formats like MPEG-2 and HEVC with various spatial/temporal resolutions.

While this is the first work to report on a polynomial regression-based approach for the prediction of HEVC PSNR values, nonetheless, we compare our solution against that reported in [13]. Again, the reviewed work is not based on machine learning and does not apply if the QP varies, hence not suitable for constant bitrate coding. Sequences 1-9 listed in Table 2 above are used in [13], hence, Table 6 presents the average correlation factor and average RMSE using these 9 sequences in comparison to the results reported in [13].

	Average RMSE (dB)	Average Corr.
Bitrate-1	0.5	0.95
Bitrate-2	0.48	0.96
Reviewed	0.6	0.98

Table 6. HEVC frame-level PSNR prediction, comparison with existing work.

The results indicate that the proposed solution is at a slight advantage in terms of RMSE and at a slight disadvantage in terms of average correlations results. Although the proposed and reviewed solutions are of different natures, nonetheless, the comparison indicates that the proposed solution is at par with state-of-the-art PSNR estimation algorithms. The extra advantage offered by the proposed solution is that it can be applied to videos with constant bitrate values. Another advantage of the proposed solution is that it can be applied at a CU level, hence the PSNR can be estimated for each and every CU in a given frame. This is important in cases where regions-based PSNR estimates are vital. One scenario is the quality assessment of video material used as a legal evidence in a court of law as mentioned in the introduction [10, 11].

The HEVC CU-level PSNR prediction results are listed in Table 7. The results show that the proposed solution can also be used to predict the PSNR at a CU level. The prediction results are not as accurate as those of the frame-level. This is so because the features available at the frame-level are numerical summaries of the CU-level features. Hence the frame-level features are much richer and better represent a video frame.

Seq. ID	RMSE (dB)	Corr.
1	1	0.94
2	0.7	0.9
3	0.8	0.85
4	1.6	0.9
5	1	0.9
6	1.1	0.86
7	1.2	0.95
8	1.2	0.88
9	1.5	0.85
10	1.0	0.91
11	0.87	0.90
12	1.16	0.83
13	0.07	0.97
14	1.3	0.93
Avg.	1.0	0.89

Table 7. HEVC CU-level PSNR prediction results

The RMSE values range from 0.7 to 1.5 dB in PSNR. While such results are acceptable, nonetheless there is still room for improvement in terms of predicting the PSNR at a CU level in HEVC videos. The HEVC results in Tables 5 and 7 for the prediction of PSNR at frame and CU levels are repeated using bitrate-2. The summaries of both predictions are presented in Table 8.

	Frame-based		CU-based	
	RMSE	Corr.	RMSE	Corr.
Bitrate-1	0.56	0.96	1.0	0.89
Bitrate-2	0.468	0.966	0.842	0.918

Table 8. Summary of HEVC PSNR prediction using two sets of bitrates.

As seen in the table, the results are very close. Yet, increasing the bitrate results in less variation in the actual PSNR values and results in slightly higher prediction accuracy.

In Figure 3, we show example histograms of the difference between the actual and predicted PSNRs at both frame and CU levels.

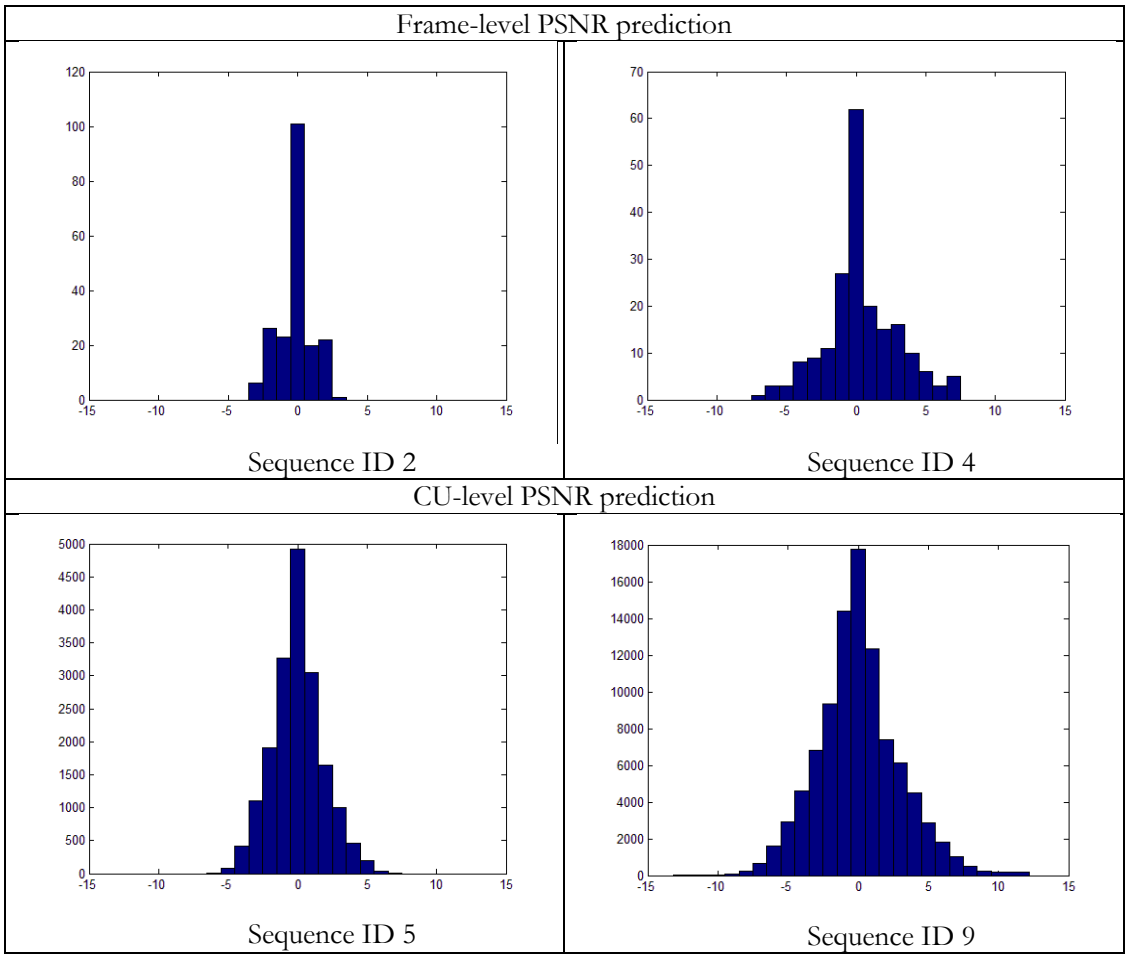

Figure 3. PSNR difference histograms for frame and CU level predictions.

In the figures, the x-axis represents the PSNR difference in dB and the y-axis is the count. The top histograms are generated for frame level prediction, whilst the lower ones are generated for CU level prediction. In the histograms displayed at the left, we show examples of good predictions and in the histograms displayed at the right, we show examples where the prediction was not as accurate. Hence the tails are longer and the standard deviation is higher.

Additionally, the CU and frame level quality predictions of HEVC are repeated using SSIM instead of PSNR. In this case the model training is based on CU and frame level SSIM values. The results are shown in Table 9.

	Frame-based		CU-based	
Seq. ID	RMSE	Corr.	RMSE	Corr.
1	0.005	0.96	0.004	0.9
2	0.002	0.97	0.001	0.88
3	0.009	0.94	0.009	0.85
4	0.004	0.95	0.005	0.93
5	0.017	0.91	0.2	0.83
6	0.002	0.9	0.004	0.96
7	0.019	0.9	0.005	0.83
8	0.005	0.94	0.02	0.9
9	0.001	0.9	0.002	0.81
10	0.007	0.94	0.005	0.87
11	0.01	0.9	0.007	0.95
12	0.002	0.91	0.002	0.8
13	0.017	0.92	0.01	0.9
14	0.007	0.88	0.003	0.89
Avg.	0.008	0.92	0.02	0.88

Table 9. HEVC CU and frame level SSIM prediction results

For the frame-based quality prediction, the average correlation factor is 0.92 and the average RMSE is 0.008, recall that SSIM values range from 0 to 1. The SSIM prediction results at the CU level are less accurate where the average correlation factor is 0.88 and the average RMSE is 0.02. This difference in accuracy is consistent with the PSNR results reported in Tables 5 and 7 above.

Again, the HEVC results in Table 8 for the prediction of SSIM at frame and CU levels are repeated using bitrate-2. The summaries of both predictions are presented in Table 10.

	Frame-based		CU-based	
	RMSE	Corr.	RMSE	Corr.
Bitrate-1	0.008	0.92	0.02	0.88
Bitrate-2	0.006	0.934	0.012	0.902

Table 10. Summary of HEVC SSIM prediction using two sets of bitrates.

As seen in the table, the results are very close. Yet, increasing the bitrate results in less variation in the actual SSIM values and results in slightly higher prediction accuracy. This was the same conclusion for the PSNR comparisons of Table 8 above.

Recently, work has been reported on the HEVC no-reference SSIM quality assessment in the existence of transmission distortions and losses [28]. Six video sequences are used and the average correlation factor reported was 76.5%. Although a direct comparison is not possible in this case, nonetheless, this gives an indication that the prediction accuracy of the proposed solution is acceptable.

Lastly, since the proposed feature extraction encompasses statistical operations, it is a good idea to compute its required processing time. The feature extraction time per video frame is reported in Table 11 for all of the spatial resolutions used.

Spatial resolution	Processing time
416x240	0.0151 s
832x480	0.0602 s
1280x720	0.1390 s
1920x1080	0.3126 s
352x288	0.0153 s

Table 11. Feature extraction time per frame using various spatial resolutions.

The feature extraction code was written using C++ and Matlab (R2013a) running on Microsoft Windows 7, 64 bits. The processer is Intel® Core™ i7 CPU @ 2.7GHz with 16 GB of RAM.

Although the code is written for research and not for production purposes, nonetheless, the results indicate that the processing times per frame are fairly reasonable.

6. Conclusion

An objective quality estimation framework is proposed in this paper. The framework is implemented using MPEG-2 and HEVC. Features are extracted from coding units and summarized to form features at frame levels. The proposed solution used stepwise regression to retain the important features and to reduce the dimensionality of the feature vectors. Thereafter, polynomial regression is used for system modeling. It was shown that the proposed system can predict the PSNR/SSIM of coding units and frames. Fourteen video sequences with various resolutions are used in the

experimental results. The average PSNR prediction correlation factor at a frame level is above 95% and the average RMSE is less than or equal to 0.6 dB. Whereas the average SSIM prediction correlation factor at a frame level is above 92% and the average RMSE is 0.008. Therefore the no-reference objective quality of HEVC coded video frames can be predicted with reasonable accuracy. At the CU level, the objective prediction results are less accurate and therefore further research can be carried out in this direction.

References:

[1] K.i Chono, Y.-Ch. Lin, D. Varodayan, Y. Miyamoto and B. Girod, "Reduced-reference image quality assessment using distributed source coding," Proc. IEEE ICME, Hannover, Germany, June, 2008.

[2] L. Yu-xin, K. Ragip and B. Udit, "Video classification for video quality prediction," Journal of Zhejiang University Science A, 7(5), pp. 919-926, 2006.

[3] G. Valenzise, S. Magni, M. Tagliasacchi and S. Tubaro, "No-Reference Pixel Video Quality Monitoring of Channel-Induced Distortion," IEEE Transactions on Circuits and Systems for Video Technology, , 22(4), pp.605,618, April, 2012.

[4] D. Schroeder, A. El Essaili, E. Steinbach, D. Staehle and M. Shehada, "Low-Complexity No-Reference PSNR Estimation for H.264/AVC Encoded Video," Proc. of 20th International Packet Video Workshop (PV), pp.1-6, 12-13 December, 2013.

[5] A. Ichigaya, M. Kurozumi, N. Hara, Y. Nishida, and E. Nakasu, "A method of estimating coding PSNR using quantized DCT coefficients", IEEE Transactions on Circuits and Systems for Video Technology, 16(2), pp. 251–259, February, 2006.

[6] T. Brandao and M.P. Queluz, "Blind PSNR estimation of video sequences using quantized DCT coefficient data," Proc. of Picture Coding Symposium, Lisbon, Portugal, November, 2007.

[7] T. Na and M. Kim, "A Novel No-Reference PSNR Estimation Method With Regard to Deblocking Filtering Effect in H.264/AVC Bitstreams," IEEE Transactions on Circuits and Systems for Video Technology, 24(2), pp.320-330, February, 2014.

[8] F. Yang and S. Wan, "Bitstream-based quality assessment for networked video: a review," IEEE Communications Magazine, 50(11), pp.203-209, November, 2012

[9] F. Yang; S. Wan; Q. Xie and H. Wu, "No-Reference Quality Assessment for Networked Video via Primary Analysis of Bit Stream," IEEE Transactions on Circuits and Systems for Video Technology, 20(11), pp.1544-1554, November, 2010.

[10] T. Shanableh, "No-Reference PSNR Identification of MPEG Video Using Spectral Regression and Reduced Model Polynomial Networks," IEEE Signal Processing Letters, 17(8), August, 2010
[11] T. Shanableh and F. Ishtiaq, "Macroblock level quality assessment using video-independent classification," Proc. of the 9th International Symposium Mechatronics and its Applications (ISMA), April, 2013.
[12] T. Shanableh, "Prediction of Structural Similarity Index of Compressed Video at a Macroblock Level," IEEE Signal Processing Letters, 18(5), May, 2011.
[13] Bumshik Lee and Munchurl Kim, "No-Reference PSNR Estimation for HEVC Encoded Video," IEEE Transactions on Broadcasting, 59(1), pp.20-27, March, 2013.
[14] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, 13(4), April, 2004.
[15] ISO/IEC 23008-2:2013, "Information technology -- High efficiency coding and media delivery in heterogeneous environments -- Part 2: High efficiency video coding," 2013.
[16] G. Sullivan, J.-R. Ohm, W.-J. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, 22(12), December, 2012.
[17] C. Wang, X. Jiang, F. Meng and Y. Wang, "Quality assessment for MPEG-2 video streams using a neural network model," in Proc. of IEEE 13th International Conference on Communication Technology (ICCT), September, 2011
[18] T. Shanableh and F. Ishtiaq, "Pattern Classification for Assessing the Quality of MPEG Surveillance Video ," in Proc. of IEEE International Conference on Computer Systems and Industrial Informatics (ICCSII'12), Sharjah, UAE, December 2012.
[19] A. Rossholm and B. Lövström, "A new low complex reference free video quality predictor," in Proc. of IEEE 10th Workshop on Multimedia Signal Processing, pp. 765-768, October, 2008.
[20] A. Rossholm and B. Lövström, "A New Video Quality Predictor based on Decoder Parameter Extraction," in Proc. of 'SIGMAP' , pp. 285-290, 2008.
[21] M. Shahid, A. Rossholm and B. Lovstrom, "A no-reference machine learning based video quality predictor," In Proc. of fifth International Workshop on Quality of Multimedia Experience (QoMEX), pp.176,181, July, 2013.
[22] M. Shahid, A. Rossholm and B. Lovstrom, "A reduced complexity no-reference artificial neural network based video quality predictor," Proc. of 4th International Congress on Image and Signal Processing (CISP), pp.517,521, October, 2011.

[23] W. Mendenhall and T. Sincich, Statistics for Engineering and Sciences, 5th edition, Pearson, 2007

[24] K.-A Toh, Q.-L. Tran and D. Srinivasan, "Benchmarking a Reduced Multivariate Polynomial Pattern Classifier," IEEE Transactions on pattern analysis and machine intelligence, 26(6), June, 2004.

[25] I.-K. Kim, K. D. McCann, K. Sugimoto, B. Bross, W.-J. Han and G. J. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 13 (HM13) Encoder Description," Document: JCTVC-O1002, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 15th Meeting: Geneva, CH, 23 Oct. – 1 November, 2013,

[26] MPEG Software Simulation Group, implementation of ISO/IEC DIS 13818-2 TM5, available online http://www.mpeg.org/MSSG/

[27] VQEG, "Final report from the video quality experts group on the validation of objective models of video quality assessment, phase II," www.vqeg.org, Tech. Rep., August, 2003.

[28] M. Abed and G. AlRegib, "No-reference quality assessment of HEVC videos in loss-prone networks," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4-9 May, 2014.

Tamer Shanableh received his Ph.D. in Electronic Systems Engineering in 2002 from the University of Essex, UK. From 1998 to 2001, he was a senior research officer at the University of Essex, during which, he collaborated with BTexact on inventing video transcoders. He joined Motorola UK Research Labs in 2001. During his affiliation with Motorola, he contributed to establishing a new profile within the ISO/IEC MPEG-4 known as the Error Resilient Simple Scalable Profile. He joined the American University of Sharjah in 2002 and is currently an associate professor of computer science. Dr. Shanableh spent the summers of 2003, 2004, 2006, 2007 and 2008 as a visiting professor at Motorola multimedia Labs. He spent the spring semester of 2012 as a visiting academic at the Multimedia and Computer Vision and Lab at the School of Electronic Engineering and Computer Science, Queen Mary, University of London, London, U.K . His research interests include digital video processing and pattern recognition.