A No-Reference Bitstream-Based Perceptual Model for Video Quality Estimation of Videos Affected by Coding Artifacts and Packet Losses

K. Pandremmenou[a], M. Shahid[b], L. P. Kondi[a], B. Lövström[b]

[a]Department of Computer Science and Engineering, University of Ioannina, GR-45110, Ioannina, Greece;
[b]Blekinge Institute of Technology, Karlskrona, Sweden

ABSTRACT

In this work, we propose a No-Reference (NR) bitstream-based model for predicting the quality of H.264/AVC video sequences, affected by both compression artifacts and transmission impairments. The proposed model is based on a feature extraction procedure, where a large number of features are calculated from the packet-loss impaired bitstream. Many of the features are firstly proposed in this work, and the specific set of the features as a whole is applied for the first time for making NR video quality predictions. All feature observations are taken as input to the Least Absolute Shrinkage and Selection Operator (LASSO) regression method. LASSO indicates the most important features, and using only them, it is possible to estimate the Mean Opinion Score (MOS) with high accuracy. Indicatively, we point out that only 13 features are able to produce a Pearson Correlation Coefficient of 0.92 with the MOS. Interestingly, the performance statistics we computed in order to assess our method for predicting the Structural Similarity Index and the Video Quality Metric are equally good. Thus, the obtained experimental results verified the suitability of the features selected by LASSO as well as the ability of LASSO in making accurate predictions through sparse modeling.

Keywords: LASSO, MOS, No-Reference, packet-loss video, quality estimation.

1. INTRODUCTION

As video communications become increasingly popular, Video Quality Assessment (VQA) gains more and more attention of the researchers. It is generally admitted that the human observer is the most reliable source for VQA. However, the collection of video subjective scores implicates a series of constraints. In subjective quality assessment tests, a number of human subjects are required to rate the video quality of the presented content. Such tests have to be carefully designed and performed and require a significant number of viewers available to perform the specific task.[1] An alternative approach to subjective tests is crowdsourcing,[2] where the testing procedure is conducted through the Internet. By following this method, one can access a wider range of evaluators, while keeping the financial cost low and obtaining results similar to those of lab-based subjective tests. Even in such a case, these tests are time-consuming and cannot be used in real-time applications.

In order to avoid subjective VQA tests, many modern methods for perceptual quality assessment have been developed and can be grouped into three categories. Full-Reference (FR) methods have access to both the original and impaired videos, and they are mostly suitable for offline applications, due to their dependence on the original video. Reduced-Reference (RR) methods use only partial information from the original video, which is transmitted to the receiver either using an ancillary channel[3] or by watermarking.[4] However, the cost of maintaining an ancillary channel may be high, while such methods may be unable to meet the requirements of quality estimation in the event of a failure in RR information delivery. No-Reference (NR) methods have access only to the impaired video, and thus, they are the most broadly applicable in real-time applications,[5] though quality estimation with limited available input information can be challenging.

E-mail: apandrem@cs.uoi.gr (K. Pandremmenou), muhammad.shahid@ieee.org (M. Shahid), lkon@cs.uoi.gr (L. P. Kondi), benny.lovstrom@bth.se (B. Lövström).

It is highly desirable to be able to estimate and control video quality automatically so as to satisfy the needs for high end-to-end Quality of Experience (QoE). Thus, objective metrics have greater potential, especially for real-time quality monitoring of video communication systems. In general, degradation of perceptual video quality can incur due to lossy video encoding and transmission losses. Lossy encoding methods are applied to video sequences in order to reduce the bit rate required for storage and/or transmission, while network congestion may result in packet losses with visible impairments on the decoded video. Therefore, there is the need for the development of a perceptual video quality model able to estimate possible video degradations due to both sources of distortion.

In the literature, several works have been proposed to address this issue. However, some of them propose FR[6] or RR[7] models, and/or they are not assessed using subjective measurements.[6] On the contrary, there are also some approaches that suggest the use of NR models for the same purpose of video quality estimation. In Ref. 8 the authors proposed two NR bitstream-based video quality metrics that use the bilinear Partial Least Squares Regression (PLSR) method and the trilinear PLSR method, where a number of features related to the properties of the encoded video sequence are extracted from the H.264/Advanced Video Coding (AVC) bitstream. In Ref. 9 a metric for video compression artifacts using supervised learning was proposed and in Ref. 10 the goal was a NR model that predicts video quality by taking into account only compression-related artifacts, proposing the use of a kernel-based learning algorithm for regression. However, Refs. 8–10 focused solely on compression artifacts and did not take into account the network-related impairments.

Clearly, the distortion due to lossy compression has a significantly different impact on perceptual video quality compared to packet-loss affected sequences. A NR H.264/AVC objective video quality metric that classifies packet losses as visible or invisible was proposed in Ref. 11. Similarly, Ref. 12 proposed a NR model that predicts continuous estimates of the visibility of packet losses in standard definition and high definition H.264/AVC video sequences. Nonetheless, both Refs. 11, 12 focused on the impact that a packet loss incurs to perceptual video quality, while compression distortion was neglected.

A work that measures the overall quality degradation due to both compression and packet losses was presented in Ref. 6. However, the specific article evaluates the severity of a packet loss in terms of Peak Signal to Noise Ratio (PSNR), which does not correlate well with human perception, and the proposed metric is of FR type, which is rather unsuitable for VQA in bandwidth-limited channels. Another research effort of the same authors[13] presented two Just Noticeable Difference (JND)-based metrics that consider both compression and network distortions. Similarly to Ref. 6, the aforementioned metrics are also of FR type. In Ref. 14, the authors developed models for predicting the objective Video Quality Metric (VQM) scores using features, which can be extracted from individual packets. However, this model was not tested for its correlation with subjective assessment.

Moreover, a NR quality assessment framework for monitoring the quality of networked video was presented in Ref. 15, where video quality measurements took place in the spatial and temporal domains. Nonetheless, this work was based on a number of assumptions and approximations, where video sequences of low resolution were considered in combination with a not very efficient Group Of Pictures (GOP) structure. In addition, the mathematical equations used in Ref. 15 were targeted at capturing the various video impairments at frame level, rather than extracting specific features able to do this task. Furthermore, in Ref. 16 a NR video quality metric was proposed that exploits the MacroBlocks (MBs) in inter-frame encoding frames under wireless networks. This metric took into account spatiotemporal characteristics by using the characters of block-based coding standards. Although the specific metric is well correlated with Mean Opinion Score (MOS), the impacts from intra-frame encoding characters are not considered by the proposed model.

The present work moves beyond the works existing in the literature and extends our previous work,[10] proposing a NR model for estimating the quality of H.264/AVC video sequences, affected by both compression artifacts and packet losses. The concept is based on the extraction of a large set of quality-relevant features from impaired bitstreams. The features extracted in this study represent virtually all of the relevant features proposed in the literature, aiming at capturing the impact of various distortion types on perceptual quality. The proposed set of features as a whole was not used before and even some features are firstly proposed in this article. The feature observations are given as input to the Least Absolute Shrinkage and Selection Operator (LASSO) regression

method,[17,18] in order for the latter to indicate those features that have the strongest effects towards video quality, and using only them to produce video quality estimates. To the best of our knowledge, this is the first time that LASSO regression is applied in order to achieve this dual goal in the building of a NR model. Additionally, in order to evaluate the efficiency of LASSO in terms of both model's sparsity and estimation accuracy, we also employed the Ordinary Least-Squares (OLS) regression method[19] as a baseline.

Our proposed model estimates the subjective ratings of the tested video sequences, that is the MOS, as well as the Structural Similarity Index (SSIM)[20] and VQM,[21] which are known for their good correlation with subjective assessment. Moreover, in order to assess the performance of our model, we emphasize on measuring the estimation accuracy, monotonicity, consistency and error. The provided experimental results signify the validity of the extracted features as well as the suitability of the features selected by LASSO in capturing the various types of video impairments, and thus, promoting efficient video quality estimations. Interestingly, we get impressive performance statistics with regard to MOS, SSIM and VQM through the use of a sparse regression model. In fact, LASSO outperforms OLS since it not only offers improved performance statistics, but also this goal is achieved through the exploitation of fewer features, meaning that less complexity is involved.

The rest of the article is structured as follows: Section 2 describes the features extracted from the bitstream in order to be used for making video quality estimations. Section 3 discusses linear regression and analyzes the proposed method as well as the OLS method, which was used for comparison purposes. Section 4 presents the employed video database along with experimental results of our method, accompanied by their interpretation. Last, Section 5 summarizes the key concepts of this work.

2. FEATURES MODELING VIDEO IMPAIRMENTS

NR models have a limited source of input information compared to FR or RR models. Due to this, we extract a large set of bitstream-based features that are expected to affect perceptual video quality in an effort to enhance the accuracy of the estimations. These features, based on how they relate to the different types of distortion, can be characterized as factors related to video content characteristics, signal factors, error factors, motion factors, and factors related to the effectiveness of the error concealment technique. Particularly, they can be described as follows:

1) **Intra**[%][10] is the percentage of I coded MBs in a slice.
2) **I4** $\times$ **4inIslice**[%][10] is the percentage of MBs of size 4×4 in an I slice.
3) **I16** $\times$ **16inIslice**[%][10] is the percentage of MBs of size 16×16 in an I slice.
4) **IinPslice**[%][10] is the percentage of I coded MBs in a P slice.
5) **P**[%][10] is the percentage of P coded MBs in a slice.
6) **PSkip**[%][10] is the percentage of P MBs coded as PSkip in a slice.
7) **P16** $\times$ **16**[%][10] is the percentage of P MBs coded with no sub-partition of MBs in a slice.
8) **P8** $\times$ **16**[%][10] is the percentage of P MBs coded with 8×16 and 16×8 partition of MBs in a slice.
9) **P8** $\times$ **8**[%][10] is the percentage of P MBs coded with 8×8 partition of MBs in a slice.
10) **P8** $\times$ **8Sub**[%][10] is the percentage of P MBs coded with 8×8 in a sub-partition of MBs in a slice.
11) **P4** $\times$ **8**[%][10] is the percentage of P MBs coded with 4×8 and 8×4 sub-partition of MBs in a slice.
12) **P4** $\times$ **4**[%][10] is the percentage of P MBs coded with 4×4 sub-partition of MBs in a slice.

13) - 20) **B_modes** correspond to the same features as given in features 5 to 12, but for B MBs.

21) - 22) $\mathbf{\Delta MV_x}$, $\mathbf{\Delta MV_y}$[10] are the average measures of motion vector difference values for x and y direction in a slice.

23) - 24) $\mathbf{avg(MV_x)}, \mathbf{avg(MV_y)}$[10] are the average measures of motion vector values for x and y directions in a slice.

25) $\mathbf{MV_0}[\%]$[10] is the percentage of motion vector values equal to zero for x and y direction in a slice.

26) $\mathbf{\Delta MV_0}[\%]$[10] is the percentage of motion vector difference values equal to zero in a slice.

27) **Motion Intensity_1**[10] is given by:

$$\sum_{i=1}^{N} \sqrt{MV_{x_i}{}^2 + MV_{y_i}{}^2}$$

where $MV_a,\ a \in [x, y]$ represents the average value of motion vectors in an MB in a-direction and N is the total number of MBs in a slice.

28) **Motion Intensity_2**[10] is given by:

$$\sqrt{avg(MV_x)^2 + avg(MV_y)^2}.$$

29) - 30) $|\mathbf{avg(MV_x)}|, |\mathbf{avg(MV_y)}|$[10] are the absolute values of average motion vector values for x and y direction in a slice.

31) **Motion Intensity_3** is given by:

$$\sum_{i=1}^{N} \sqrt{|(MV_x)|_i^2 + |(MV_y)|_i^2}$$

where $|(MV_a)|$ represents the absolute value of motion vectors in an MB in a-direction.

32) **Motion Intensity_4** is given by:

$$\sqrt{|avg(MV_x)|^2 + |avg(MV_y)|^2}.$$

33) **NotStill**[22] is a boolean variable, which is true, if the value of "Motion Intensity_2" feature is over $1/10^{\text{th}}$ of the highest magnitude value of all sequences.

34) **HighMot**[22] is a boolean variable, which is true, if the value of "Motion Intensity_2" feature is over $8/10^{\text{th}}$ of the highest magnitude value of all sequences.

35) - 36) **MaxResEngy, MeanResEngy**[23] are the maximum and mean residual energy values over all the MBs of a slice. The residual energy for an MB is computed as the sum of squares of its transform coefficients.

37) **LR** is a boolean variable, which is true, if a slice is lost.

38) **LostSinFrm** is the number of lost slices in a frame.

39) **Height**[22] is the vertical location of the lost slice within a frame.

40) **TMDR**[22] is the number of frames affected by a lost slice, due to error propagation.

41) **SpatialExtend**[22] is the number of consecutive lost slices in a frame.

42) **SpatialExtend2**[22] is a boolean variable, which is true, if SpatialExtend=2.

43) **SpatialExtendFrm**[22] is a boolean variable, which is true, if all slices of a frame are lost.

44) **Error1Frm**[22] is a boolean variable, which is true, if TMDR=1.

45) **DistToRef**[22] is the distance in frames between the current frame and the reference frame used for concealment. Based on our considered GOP pattern, P frames are concealed using images from a temporal distance of three frames ago, while both I frames and B frames are concealed using images from a temporal distance of one frame ago.

46) **FarConceal**[22] is a boolean variable, which is true, if $|\text{DistToRef}| \geq 3$.

It is worth mentioning that most of the features are inspired from Refs. 10, 22, 23. However, features 13-20 and 31, 32, 37, 38 are, to our knowledge, proposed for the first time in the current study, in order to contribute to the achievement of more precise video quality estimations. For the sequence encoding considered in the database that we used in this work,[1] a packet corresponds to a video slice, where each slice consists of a full row of MBs. In light of this, calculating feature values for each slice of the video can better model the impairments due to possible packet losses. Therefore, the features that are related to the occurrence of a packet loss are computed at slice level, while the features that are related to motion vectors, are more suitably computed at MB level. In the following, we apply a bottom-up approach and take average values for all MB-level features at slice level, and next we average further these values to obtain their corresponding values for each video sequence.

3. REGRESSION METHODS

In this section we briefly overview linear regression and in the following, we present the LASSO regression method employed to estimate video quality. Also, the OLS regression method is analyzed, which was used for comparison.

A linear regression model is a model of the form:

$$\hat{y}_i = w_0 + \sum_{j=1}^{m-1} w_j \phi_j(x_i) = w^\top \phi(x_i), \quad \text{for } i = 1, \ldots, n$$

where n is the total number of observations, namely the total number of examined slices; $\hat{y}_i$ is the estimated value of perceptual quality at observation x_i; the basis function $\phi(x_i)$ is a vector of m-by-1 values at observation x_i, which includes the values for all examined features for a particular slice, and w is an m-by-1 vector of regression coefficients including the intercept term w_0. Such a model is *linear* in the coefficients w.

3.1 Ordinary Least-Squares (OLS) Regression

Ordinary Least-Squares regression[19] is by far the most widely used method for regression because of its ease of implementation as well as its simplicity. It determines the regression coefficients w, by solving the following minimization problem:

$$\min_w \left(\frac{1}{2} \sum_{i=1}^{n} \left(y_i - w^\top \phi(x_i) \right)^2 \right) = \min_w \frac{1}{2} \|y - \Phi w\|^2. \tag{1}$$

From Eq. (1) we observe that the fitted coefficients minimize the mean squared difference between the n-by-1 vector y of measured perceptual quality values and the prediction vector $\Phi\, w$. Specifically, Φ is the n-by-m design matrix for the model, i.e., the matrix that includes the values for all examined features of all slices, and w is the vector of regression coefficients as it results from the solution of the problem of Eq. (1). Particularly, it is given by:

$$w = (\Phi^\top \Phi)^{-1} \Phi^\top y.$$

In the following, the calculated weights (regression coefficients), as they resulted from the training phase, are applied to the test data so as to get the estimated perceptual quality values $\hat{y}$.

However, there are two major disadvantages of this method that make its use problematic. The first is the prediction accuracy. There are cases where many of the features of the design matrix are highly correlated and

thus, we can get inaccurate results about any regression coefficient assigned to a predictor. Additionally, a high degree of multi-collinearity means that the matrix Φ is not of full rank and hence, neither is $\Phi^\top \Phi$. Therefore, the inversion of $\Phi^\top \Phi$ is infeasible or the results after such an inversion may be imprecise. The other major concern of OLS is that of interpretation. Having a large number of features, we often desire to select a small subset of them, by keeping the features that capture the strongest effects towards video quality. In fact, some of the computed features may be irrelevant or noise, leading to estimation harming. A method that circumvents both of these issues is the one presented below.

3.2 Least Absolute Shrinkage and Selection Operator (LASSO) Regression

Least Absolute Shrinkage and Selection Operator regression[18,24] was proposed by Tibshirani,[17] as an innovative tool that improves the estimation accuracy of ill-posed problems and can be used for both selection of regression coefficients as well as response variable estimation. It is a linear model, which is useful in some contexts due to its tendency to prefer solutions with fewer parameter values, effectively reducing the number of variables upon which the given solution is dependent, and producing interpretable models like subset selection, by exhibiting the stability of ridge regression[25] at the same time.[17]

The specific method minimizes the residual sum of squares, subject to the sum of the absolute value of the regression coefficients being less than a constant. Mathematically, for a given nonnegative λ value, it solves the following penalized least squares problem:

$$\min_{w} \left(\frac{1}{2} \sum_{i=1}^{n} \left(y_i - w^\top \phi(x_i) \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n} |w_i| \right). \tag{2}$$

From Eq. (2) we observe that the fitted coefficients minimize the mean squared difference between the measured perceptual quality values and the design matrix multiplied by the regression coefficient values. The minimization problem involves the penalization of the sum of the absolute values of the regression coefficients, where the amount of the regularization is controlled by λ. It holds that the larger the λ values are, the more coefficients w_i are driven to zero, leading in this way to a sparse model representation. Likewise, for $\lambda = 0$ no shrinkage is performed and hence, the solution of the OLS method is obtained. For our experiments, we selected the $\lambda > 0$ value that corresponds to the lowest Mean Squared Error (MSE) value of the first term of Eq. (2), for each model. Furthermore, it is to be noted that in all considered regression models, we included the intercept term, in order to absorb the bias, since we empirically observed that its inclusion greatly improved the convergence of each considered regression model.

Tables 1 and 2 include the intercept values as well as the regression coefficient values (listed from 1 to 46) assigned to each feature, for the estimation of MOS, SSIM and VQM quality metrics, when OLS and LASSO regression is applied, respectively. Also, in Table 2 the row "λ" depicts the λ values for each model, used in Eq. (2). From the provided results of these tables we can see that LASSO is a sparser approach compared to OLS. It keeps less than 1/3 of the input features, for each model, assigning zero regression coefficient values to the rest, and thus, it renders them useless. On the contrary, the OLS method assigns non zero regression coefficients to all of the features, without eliminating any possible redundancies in the input data. Therefore, it poses the risk of deteriorating the estimations of the quality values.

4. EXPERIMENTAL RESULTS

In order to evaluate the behavior of the proposed models that attempt to estimate video quality in terms of MOS, SSIM and VQM, evidently, it was necessary to calculate the true values for each of these metrics. The computation of the actual SSIM and VQM values was a trivial task, while we were supplied with the MOS values, generated at the Ecole Polytechnique Fédérale de Lausanne (EPFL) and Politecnico di Milano (PoliMi).[1] More specifically, in our experiments we dealt with the results obtained from EPFL.

The specific publicly available database[1] was formed with the goal of supporting reproducible research in the field of quality assessment algorithms. Particularly, the test material used in our experiments included the "Foreman", "Mother" and "Paris" video sequences at Common Intermediate Format (CIF) resolution (352x288

Table 1: Intercept and regression coefficient values achieved by OLS.

Features	MOS	SSIM	VQM
0) Intercept	-13.5897	0.4593	14.3036
1) Intra[%]	-0.0686	-0.0020	0.0781
2) I4 $\times$ 4inIslice[%]	0.9315	0.0101	-0.3498
3) I16 $\times$ 16inIslice[%]	-0.0643	-0.0058	-0.1963
4) IinPslice[%]	0.0294	-2.7338×10^{-4}	-0.0382
5) P[%]	0.0260	0.0017	-0.0598
6) PSkip[%]	0.0452	4.2590×10^{-4}	-0.0350
7) P16 $\times$ 16[%]	0.0409	-0.0013	-0.0194
8) P8 $\times$ 16[%]	0.2989	0.0046	-0.2508
9) P8 $\times$ 8[%]	-0.3785	-0.0022	0.2563
10) P8 $\times$ 8Sub[%]	0.1773	-0.0011	-0.0400
11) P4 $\times$ 8[%]	0.0603	-4.2319×10^{-4}	0.0913
12) P4 $\times$ 4[%]	-0.4568	-0.0087	0.5975
13) B[%]	0.1297	0.0059	-0.1203
14) BSkip[%]	0.0221	-1.3470×10^{-4}	-0.0060
15) B16 $\times$ 16[%]	-0.0141	-2.0760×10^{-5}	0.0241
16) B8 $\times$ 16[%]	-0.0548	4.7298×10^{-4}	6.8552×10^{-4}
17) B8 $\times$ 8[%]	0.0250	-7.1516×10^{-4}	0.0503
18) B8 $\times$ 8Sub[%]	-0.0788	-4.6022×10^{-4}	0.0657
19) B4 $\times$ 8[%]	0.2776	2.6528×10^{-5}	-0.0151
20) B4 $\times$ 4[%]	1.6525	-0.0020	-0.1985
21) ΔMV_x	0.3324	-0.0016	-0.1912
22) ΔMV_y	-5.7547	-0.0445	3.9846
23) avg(MV_x)	0.0627	1.8509×10^{-4}	-0.0429
24) avg(MV_y)	0.2392	1.9322×10^{-4}	-0.1718
25) MV_0[%]	25.2009	-0.0110	-10.7282
26) ΔMV_0[%]	-4.7468	-0.0447	4.9602
27) Motion Intensity_1	0.0011	7.6397×10^{-7}	-4.7568×10^{-4}
28) Motion Intensity_2	-0.0028	-1.8478×10^{-4}	-0.0107
29) |avg(MV_x)|	-0.0014	-2.2317×10^{-4}	3.5045×10^{-4}
30) |avg(MV_y)|	0.0036	-7.2850×10^{-4}	-0.0363
31) Motion Intensity_3	-0.0015	-9.3395×10^{-6}	4.7530×10^{-4}
32) Motion Intensity_4	-0.0337	-3.5414×10^{-4}	0.0187
33) NotStill	-430.7387	-6.8888	885.7327
34) HighMot	-93.6475	-1.4840	223.0036
35) MaxResEngy	697.6703	12.1826	-1.2948×10^3
36) MeanResEngy	-8.9787	0.0533	1.3253
37) LR	17.7861	-0.1709	0.3835
38) LostSinFrm	-8.5533	0.1025	-29.0802
39) Height	15.0957	-0.7480	-5.6862
40) TMDR	119.6991	-1.0521	131.5921
41) SpatialExtend	2.6052	0.8778	3.1276
42) SpatialExtend2	425.1855	11.9312	-1.1407×10^3
43) SpatialExtendFrm	3.8337×10^{-9}	8.4789×10^{-11}	-2.4944×10^{-8}
44) Error1Frm	-5.7420×10^{-9}	-1.5256×10^{-12}	3.1181×10^{-9}
45) DistToRef	0.8024	-0.0284	-0.0715
46) FarConceal	83.2314	-3.0833	129.9600

pixels) and "Ice", "Harbour" and "Parkjoy" video sequences at 4CIF resolution (704x576 pixels). All CIF sequences as well as "Harbour" consisted of 298 frames, while "Ice" and "Parkjoy" comprised of 238 and 250

Table 2: Intercept and regression coefficient values achieved by LASSO.

Features	MOS	SSIM	VQM
0) Intercept	2.6052	0.0832	-0.1886
1) Intra[%]	**0.0000**	**0.0000**	**0.0000**
2) I4 $\times$ 4inIslice[%]	**0.0000**	2.1534×10^{-4}	0.2482
3) I16 $\times$ 16inIslice[%]	-0.0831	-0.0044	**0.0000**
4) IinPslice[%]	**0.0000**	**0.0000**	**0.0000**
5) P[%]	**0.0000**	**0.0000**	**0.0000**
6) PSkip[%]	**0.0000**	**0.0000**	**0.0000**
7) P16 $\times$ 16[%]	**0.0000**	**0.0000**	**0.0000**
8) P8 $\times$ 16[%]	**0.0000**	0.0047	**0.0000**
9) P8 $\times$ 8[%]	**0.0000**	**0.0000**	**0.0000**
10) P8 $\times$ 8Sub[%]	**0.0000**	**0.0000**	**0.0000**
11) P4 $\times$ 8[%]	**0.0000**	**0.0000**	**0.0000**
12) P4 $\times$ 4[%]	**0.0000**	**0.0000**	1.9112
13) B[%]	**0.0000**	0.0124	**0.0000**
14) BSkip[%]	**0.0000**	**0.0000**	**0.0000**
15) B16 $\times$ 16[%]	**0.0000**	**0.0000**	**0.0000**
16) B8 $\times$ 16[%]	**0.0000**	**0.0000**	**0.0000**
17) B8 $\times$ 8[%]	**0.0000**	**0.0000**	**0.0000**
18) B8 $\times$ 8Sub[%]	**0.0000**	**0.0000**	**0.0000**
19) B4 $\times$ 8[%]	**0.0000**	**0.0000**	**0.0000**
20) B4 $\times$ 4[%]	**0.0000**	**0.0000**	**0.0000**
21) $\Delta \mathrm{MV_x}$	**0.0000**	**0.0000**	**0.0000**
22) $\Delta \mathrm{MV_y}$	**0.0000**	**0.0000**	2.7188
23) avg($\mathrm{MV_x}$)	**0.0000**	**0.0000**	**0.0000**
24) avg($\mathrm{MV_y}$)	**0.0000**	**0.0000**	**0.0000**
25) $\mathrm{MV_0}$[%]	**0.0000**	**0.0000**	**0.0000**
26) $\Delta \mathrm{MV_0}$[%]	2.4405	**0.0000**	2.1070
27) Motion Intensity_1	**0.0000**	**0.0000**	**0.0000**
28) Motion Intensity_2	**0.0000**	-1.3768×10^{-6}	**0.0000**
29) $\lvert\mathrm{avg}(\mathrm{MV_x})\rvert$	**0.0000**	**0.0000**	**0.0000**
30) $\lvert\mathrm{avg}(\mathrm{MV_y})\rvert$	**0.0000**	**0.0000**	**0.0000**
31) Motion Intensity_3	**0.0000**	**0.0000**	**0.0000**
32) Motion Intensity_4	**0.0000**	**0.0000**	**0.0000**
33) NotStill	-1.1180	-0.2925	**0.0000**
34) HighMot	**0.0000**	**0.0000**	**0.0000**
35) MaxResEngy	3.0031×10^{-9}	**0.0000**	**0.0000**
36) MeanResEngy	-7.3627×10^{-9}	-7.6440×10^{-11}	-2.8375×10^{-8}
37) LR	-400.7000	-0.1382	951.9403
38) LostSinFrm	-8.1379	**0.0000**	1.3923
39) Height	16.4351	-0.0525	**0.0000**
40) TMDR	-14.3535	-0.5829	-19.2934
41) SpatialExtend	13.9946	-0.2504	-3.8692
42) SpatialExtend2	115.6684	1.6991	109.4864
43) SpatialExtendFrm	**0.0000**	**0.0000**	**0.0000**
44) Error1Frm	325.8795	1.8654	-985.2688
45) DistToRef	**0.0000**	0.0604	**0.0000**
46) FarConceal	452.2033	5.2607	-792.4988
λ	9.9335×10^{-5}	4.3801×10^{-6}	1.2340×10^{-4}

frames, respectively. The GOP structure was IBBP with a GOP size of 16 frames. The sources were encoded using the JM 14.2 version of H.264/AVC reference software using the High profile, where a full row of MBs was coded as a separate slice and each of the sequences was corrupted with a Packet Loss Rate (PLR) of $0.1\%, 0.4\%, 1\%, 3\%$ and 5%. The video sequences used for the model training were the "Foreman", "Mother", "Paris", "Harbour" and "Parkjoy", for all PLRs considered in this work, and the "Ice" video sequences for all different PLRs were used for testing the performance of our model.

According to VQEG Phase I report on the validation of reduced-reference and no-reference objective models for standard definition television,[26] the performance of a quality estimation model can be evaluated by the parameters:

1. **Pearson Correlation Coefficient** (PCC), which measures the linear relationship between estimated and measured video quality values. It is given by:

$$\mathrm{PCC} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \tilde{y})\,(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \tilde{y})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3}$$

   where $\hat{y}_i, y_i$ represent the estimated and measured video quality values, respectively; $\tilde{y}, \bar{y}$ represent the mean of the estimated and measured video quality values, respectively, and n is the total number of each such value. It holds that PCC values close to 0 declare bad or no correlation and values close to 1 denote high positive correlation.[27]

2. **Root Mean Squared Error** (RMSE), which evaluates the accuracy of the proposed model, by calculating the difference between the estimated and measured quality values. It is given by:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y)^2}.$$

   However, in this paper we selected to normalize the RMSE values, since MOS, SSIM and VQM scores have different scales. Hence, we computed the **Normalized RMSE** (NRMSE), by dividing RMSE with the range of the estimated scores of each quality metric, respectively:[28]

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{\hat{y}_{max} - \hat{y}_{min}}. \tag{4}$$

3. **Outlier Ratio** (OR), which measures the consistency of an objective metric and is expressed as the ratio of the number of outlier points to the total number of data points.

$$\mathrm{OR} = \frac{\text{Number of outliers}}{\text{Total number of data points}}$$

   where

$$\text{Number of outliers} : |perr(i)| > k2\,\frac{stdDMOS(i)}{\sqrt{Nsubj}}. \tag{5}$$

   The amount $perr(i)$ is the estimation error between the corresponding estimated and measured value of a video sequence i. The constant $k2$ is equal to 1.96 to account for 95% confidence interval, $stdDMOS(i)$ is the standard deviation of the individual scores associated with a video sequence i, and $Nsubj$ is the number of viewers per video sequence i. In this work, 16 viewers evaluated the CIF video sequences and 17 viewers evaluated the 4CIF video sequences, after outliers removal on the results collected by EPFL.[1] Thus, a data point is considered as an outlier if the absolute value of $perr(i)$ is greater than the right term of Rel. (5).

   Furthermore, we examined two additional measures of performance that offer information about the monotonicity and the error of the estimations in relation with the measured values. Particularly,

4. **Spearman Rank Order Correlation Coefficient** (SROCC)[27] is related to the monotonicity between the estimated and measured video quality and is given by:

$$\text{SROCC} = 1 - \frac{6\sum_{i=1}^{n} {d_i}^2}{n(n^2-1)} \tag{6}$$

where d_i denotes the difference in ranks between each pair of estimated and measured video quality values and n is the total number of each such value. It holds that $-1 \leq \text{SROCC} \leq 1$, where SROCC $= 1$ denotes a perfect positive Spearman correlation, with the estimated values being a perfect monotone function of the measured values and vice versa for SROCC $= -1$.

5. **Mean Absolute Error** (MAE), which is given by:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y|. \tag{7}$$

The performance results of our proposed models in terms of the indices described above in the database of Ref. 1 are tabulated in Tables 3 and 4, when OLS and LASSO regression is applied, respectively. In these tables, the row "Number of Used Features" presents the number of features used by each regression technique for making quality estimations; that is the number of features that were assigned a non zero regression coefficient value.

Table 3: Performance results for "Ice" using OLS.

Performance Statistics	**MOS**	**SSIM**	**VQM**
PCC	0.9152	0.9992	0.8786
NRMSE	1.2627	3.7919	0.4312
OR	0.2000	0.2000	0.0000
SROCC	1.0000	0.9000	1.0000
MAE	4.4618	0.1141	1.5382
Number of Used Features	46	46	46

Table 4: Performance results for "Ice" using LASSO.

Performance Statistics	**MOS**	**SSIM**	**VQM**
PCC	0.9173	0.9982	0.8800
NRMSE	0.1476	3.1699	0.2720
OR	0.0000	0.2000	0.0000
SROCC	1.0000	1.0000	1.0000
MAE	0.4287	0.0951	0.8066
Number of Used Features	13	15	12

A close inspection of the results reveals that the performance statistics achieved using the LASSO regression method are improved compared to the results achieved by OLS. The difference between the results of the two methods is more obvious in terms of the NRMSE and MAE measures. With only an isolated exception for the PCC of SSIM, where OLS offers marginally better estimation accuracy, in all other cases the results of Table 4 highlight the ability of LASSO in producing precise quality estimates, carefully selecting the most influential features towards video quality, at the same time. Compared to OLS, a significantly reduced number of features is used, contributing in this way to the reduction of the problem's complexity. Hereinafter, only the specific feature subset can be used to estimate video quality in any real-time application.

5. CONCLUSIONS

In this work, we develop a NR video quality model, which accounts for the impact of compression artifacts as well as the impairments due to possible packet losses. A large set of features is extracted from the impaired bitstreams

and the LASSO regression model is used to perform feature selection in order to reject the features that harm the quality estimates, and produce video quality estimations that correlate well with subjective assessment. For comparison purposes, we utilize the OLS regression method and except for MOS, we build models able to estimate SSIM and VQM. The experimental results signify that LASSO achieves high performance statistics using only a few features, as opposed to OLS. Despite the fact that OLS presents a competitive (but worse) performance with LASSO in terms of the examined measures of performance, it employs a much larger number of features increasing problem's complexity. Moreover, the validity of the features used by LASSO lead to a very good correlation of our model with MOS, SSIM and VQM. Hence, LASSO is a good choice for building a simple and low-complexity model that offers high correlation with subjective ratings and FR metrics, when it is supplied with appropriate, quality-relevant features.

ACKNOWLEDGEMENT

Effort sponsored by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-12-1-0001. The U.S Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright notation thereon.

REFERENCES

[1] De Simone, F., Naccari, M., Tagliasacchi, M., Dufaux, F., Tubaro, S., and Ebrahimi, T., "Subjective quality assessment of H.264/AVC video streaming with packet losses," *EURASIP Journal on Image and Video Processing* **2011 Article ID 190431** (2011).

[2] Keimel, C., Habigt, J., and Diepold, K., "Challenges in crowd-based video quality assessment," in [*Fourth International Workshop on Quality of Multimedia Experience (QoMEX)*], 13–18 (Jul. 2012).

[3] Pinson, M. H. and Wolf, S., "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting* **50**, 312–322 (Sept. 2004).

[4] Zeng, K. and Wang, Z., "Quality-aware video based on robust embedding of intra- and inter-frame reduced-reference features," in [*17th IEEE International Conference on Image Processing (ICIP)*], 3229–3232 (Sept. 2010).

[5] Reibman, A. R., Vaishampayan, V. A., and Sermadevi, Y., "Quality monitoring of video over a packet network," *IEEE Transactions on Multimedia* **6**, 327–334 (Apr. 2004).

[6] Liu, T., Wang, Y., Boyce, J. M., Yang, H., and Wu, Z., "A novel video quality metric for low bit-rate video considering both coding and packet-loss artifacts," *IEEE Journal of Selected Topics in Signal Processing* **3**, 280–293 (Apr. 2009).

[7] Masry, M., Hemami, S. S., and Sermadevi, Y., "A scalable wavelet-based video distortion metric and applications," *IEEE Transactions on Circuits and Systems for Video Technology* **16**, 260–273 (Feb. 2006).

[8] Keimel, C., Habigt, J., Klimpke, M., and Diepold, K., "Design of no-reference video quality metrics with multiway partial least squares regression," in [*Third International Workshop on Quality of Multimedia Experience (QoMEX)*], 49–54 (Sept. 2011).

[9] Vink, J. P. and de Haan, G., "No-reference metric design with machine learning for local video compression artifact level," *IEEE Journal of Selected Topics in Signal Processing* **5**, 297–308 (Apr. 2011).

[10] Shahid, M., Rossholm, A., and Lövström, B., "A no-reference machine learning based video quality predictor," in [*Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*], 176–181 (Jul. 2013).

[11] Staelens, N., Vercammen, N., Dhondt, Y., Vermeulen, B., Lambert, P., Van de Walle, R., and Demeester, P., "Viqid: A no-reference bit stream-based visual quality impairment detector," in [*Second International Workshop on Quality of Multimedia Experience (QoMEX)*], 206–211 (Jun. 2010).

[12] Argyropoulos, S., Raake, A., Garcia, M.-N., and List, P., "No-reference video quality assessment for SD and HD H.264/AVC sequences based on continuous estimates of packet loss visibility," in [*Third International Workshop on Quality of Multimedia Experience (QoMEX)*], 31–36 (Sept. 2011).

[13] Liu, T., Yang, H., Stein, A., and Wang, Y., "Perceptual quality measurement of video frames affected by both packet losses and coding artifacts," in [*First International Workshop on Quality of Multimedia Experience, (QoMEX)*], 210–215 (Jul. 2009).

[14] Wang, C., Lin, T.-L., and Cosman, P. C., "Network-based model for video packet importance considering both compression artifacts and packet losses," in [*IEEE Global Telecommunications Conference (GLOBECOM)*], 1–5 (Dec. 2010).

[15] Yang, F., Wan, S., Xie, Q., and Wu, H. R., "No-reference quality assessment for networked video via primary analysis of bit stream," *IEEE Transactions on Circuits and Systems for Video Technology* **20**, 1544–1554 (Nov. 2010).

[16] Yang, Y., Wen, X., Zheng, W., Yan, L., and Zhang, A., "A no-reference video quality metric by using inter-frame encoding characters," in [*14th International Symposium on Wireless Personal Multimedia Communications (WPMC)*], 1–5 (Oct. 2011).

[17] Tibshirani, R., "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society, Series B* **58**, 267–288 (1994).

[18] Tibshirani, R., "The LASSO method for variable selection in the Cox model," *Statist. Med.* **16**(4), 385–395 (1997).

[19] Bishop, C. M., [*Pattern Recognition and Machine Learning (Information Science and Statistics)*], Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006).

[20] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing* **13**, 600–612 (Apr. 2004).

[21] National Telecommunications & Information Administration, "Video quality metric (VQM) software." http://www.its.bldrdoc.gov/resources/video-quality-research/software.aspx.

[22] Lin, T.-L., Kanumuri, S., Zhi, Y., Poole, D., Cosman, P. C., and Reibman, A. R., "A versatile model for packet loss visibility and its application to packet prioritization," *IEEE Transactions on Image Processing* **19**, 722–735 (Mar. 2010).

[23] Kanumuri, S., Subramanian, S. G., Cosman, P. C., and Reibman, A. R., "Predicting H.264 packet loss visibility using a generalized linear model," in [*IEEE International Conference on Image Processing (ICIP)*], 2245–2248 (Oct. 2006).

[24] Osborne, M. R., Presnell, B., and Turlach, B. A., "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis* **20**(3), 389–404 (2000).

[25] Marquardt, D. W. and Snee, R. D., "Ridge regression in practice," *The American Statistician* **29**, 3–20 (Feb. 1975).

[26] VQEG, "Final report from the Video Quality Experts Group on the validation of reduced-reference and no-reference objective models for standard definition television, Phase I," (Jun. 2009).

[27] Hinkle, D. E., Wiersma, W., and Jurs, S. G., [*Applied statistics for the behavioral sciences.*], Boston, Mass: Houghton Mifflin (2003).

[28] Shanableh, T., "Prediction of structural similarity index of compressed video at a macroblock level," *IEEE Signal Processing Letters* **18**, 335–338 (May 2011).