DESIGN OF NO-REFERENCE VIDEO QUALITY METRICS WITH MULTIWAY PARTIAL LEAST SQUARES REGRESSION

Christian Keimel, Julian Habigt, Manuel Klimpke and Klaus Diepold

Technische Universität München, Institute for Data Processing,
Arcisstr. 21, 80333 Munich, Germany
christian.keimel@tum.de, jh@tum.de, msk@mytum.de, kldi@tum.de

ABSTRACT

No-reference video quality metrics are becoming ever more popular, as they are more useful in real-life applications compared to full-reference metrics. One way to design such metrics is by applying data analysis methods on both objectively measurable features and data from subjective testing. Partial least squares regression (PLSR) is one such method. In order to apply such methods, however, we have to temporally pool over all frames of a video, loosing valuable information about the quality variation over time. Hence, we extend the PLSR into a higher dimensional space with multiway PLSR in this contribution and thus consider video in all its dimensions. We designed a H.264/AVC bitstream no-reference video quality metric in order to verify multiway PLSR against PLSR with respect to the prediction performance. Our results show that the inclusion of the temporal dimension with multiway PLSR improves the quality prediction and its correlation with the actual quality.

Index Terms— H.264/AVC, video quality metric, no-reference metric, multilinear data analysis, multiway PLSR, trilinear PLS.

1. INTRODUCTION

In the traditional approach to video quality metrics, the goal is to build a model of the spatial and temporal properties of the human visual system (HVS) as well as possible in order to predict quality perception of human observers adequately. This, however, assumes that the HVS is understood well enough to create a comprehensive model. If we utilize a more data driven approach, we consider the HVS as a black box: we use data analysis methods to determine the relationship between objectively measurable features at its input and the subjective quality at the box's output. Partial Least Squares Regression (PLSR) is used in some of our previous contributions [1, 2].

The temporal nature of video, however, is often neglected. Features are determined on a frame-by-frame basis and then temporally pooled over all frames. The data analysis is then performed only on these pooled features. This arbitrary pooling, especially averaging, obscures the influence of temporal distortions on the quality perception and thus leads to less than optimal models as shown in [2].

We therefore propose in this contribution to extend the data analysis with PLSR into the higher dimensional space with multiway PLSR for the design of video quality metrics. In doing so, we can avoid temporal pooling and build our model directly with the three dimensional video cube. To demonstrate the advantage of the multidimensional approach, we will design no-reference video quality metrics based on features extracted from encoded H.264/AVC bitstreams for HDTV similar to [3] and compare the results to PLSR
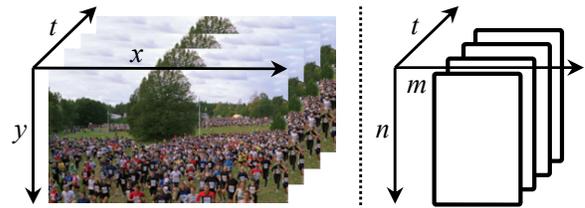
Fig. 1: Video cube of a video sequence with t frames at a spatial resolution of $x \times y$ pixels (left) and corresponding m features for n different video sequences with t frames.

with temporally pooled features. Such a metric is of course limited in its scope due to the necessary adaption to a certain coding technology and its bitstream format, but considering that H.264/AVC is the predominant coding technology for HDTV, the scope can be assumed to be wide enough. To the best of our knowledge, this is the first use of multiway PLSR in the design of video quality metrics.

PLSR itself has already been used for the design of video quality metrics in [1, 2]. A video quality metric with the multi-dimensional extension of the principal component regression (PCR), 2D-PCR, was presented in [4]. In related contributions to bitstream feature no-reference video quality metrics, Eden estimates the PSNR of interlaced HDTV video sequences with H.264/AVC bitstream features in [5] whereas Slanina et al. in [6] estimate the PSNR for videos in CIF resolution. Rossholm and Lövström not only estimate PSNR in [7], but also other video quality metrics for videos in CIF resolution from the bitstream. In [8], Lee et al. use bitrate, QP and deblocking filter parameters for quality prediction of QCIF resolution videos, but no different coding structures were considered in the contribution by Lee et al. Another approach is the combination of bitstream features and features extracted from the decoded video sequences in a hybrid metric as proposed for interlaced HDTV by Sugimoto et al. in [9].

This contribution is organized as follows: firstly, we will discuss PLSR and multiway PLSR, before introducing the feature extraction. We then shortly discuss the model building, followed by a description of the subjective testing. After presenting and discussing the results, we will conclude with a short summary.

2. DESIGN OF VIDEO QUALITY METRICS WITH PARTIAL LEAST SQUARES REGRESSION

In the data analysis approach, we do not assume a-priori specific relationships between the features and the visual quality, but rather gain the relationships by analyzing the available data. Firstly, we construct a data matrix $\mathbf{X}$ where the rows correspond to data from individual sequences and the columns represent the features. The visual quality values that were determined in subjective tests are represented by the $n \times 1$ column vector $\mathbf{y}$. With n sequences and m features, $\mathbf{X}$ is an $n \times m$ matrix. Our aim is to find the unknown $m \times 1$ regression weight vector $\mathbf{b}$, mapping the features to the visual quality

$$\mathbf{y} = \mathbf{Xb}. \tag{1}$$

Note that we collapsed the temporal dimension of the $n \times m \times t$ video cube by temporal pooling as shown in Fig. 1 into a $1 \times m$ row vector along t.

Although we extracted a large number of parameters, not all will be useful in predicting the visual quality of a video sequence, as we will see shortly. A principal overview of the model building with data analysis is given in Fig. 2.

2.1. Bilinear Partial Least Squares Regression

PLSR is an extension of the principal component regression method (PCR). For PCR, the data matrix $\mathbf{X}$ is first subjected to a PCA, and then for selected principal components (PC) a regression on $\mathbf{y}$ is done. The disadvantage of PCR is that the PCs best suited to represent $\mathbf{X}$, carrying the structure of the videos, are not necessarily the same PCs best suited to explain the variance in $\mathbf{y}$, describing the quality variation of the videos. In contrast, the modeling with PLSR is done simultaneously on $\mathbf{X}$ and $\mathbf{y}$, ensuring PCs that explain the variance in both $\mathbf{X}$ and $\mathbf{y}$ best. This basic type of PLSR is also called bilinear partial least squares or PLS1 . Bilinear relates to the two-dimensional nature of the data matrix $\mathbf{X}$, and 1 denotes that the dependent variable $\mathbf{y}$ is a vector. The iterative algorithm shown in Listing 1 describes how we can extract the first g PCs with PLS1.

Algorithm 1: Bilinear PLS1

center $\mathbf{X}$ and $\mathbf{y}$
$\mathbf{X}_1 = \mathbf{X}$ and $\mathbf{y}_1 = \mathbf{y}$
$f = 1$

1 $\mathbf{w}_f = \dfrac{\mathbf{X}_f^T \mathbf{y}_f}{\|\mathbf{X}_f^T \mathbf{y}_f\|}$

2 $\mathbf{t}_f = \mathbf{X}_f \mathbf{w}_f$

3 $\hat{q}_f = \dfrac{\mathbf{t}_f^T \mathbf{y}_f}{\mathbf{t}_f^T \mathbf{y}_f}$ and $\mathbf{p}_f = \dfrac{\mathbf{X}_f^T \mathbf{y}_f}{\mathbf{t}_f^T \mathbf{y}_f}$

4 Let $\mathbf{X}_{f+1} = \mathbf{X}_f - \mathbf{t}_f \mathbf{p}_f^T$ and $\mathbf{y}_{f+1} = \mathbf{y}_f - \mathbf{t}_f \hat{q}_f$

5 Continue from 1 and let $f = f + 1$ until $f = g$

Based on the extracted PCs $\mathbf{w}$, the scores $\mathbf{t}$ and loadings $\mathbf{p}$, we can then obtain an estimation $\hat{\mathbf{b}}$ of the regression weight vector $\mathbf{b}$ and thus can write the quality estimation $\hat{\mathbf{y}}$ for $\mathbf{y}$ as

$$\hat{\mathbf{y}} = \mathbf{1}\hat{b}_0 + \mathbf{X}\hat{\mathbf{b}} + \mathbf{e}, \tag{2}$$

where $\hat{b}_0$ describes the offset and $\mathbf{e}$ the estimation error of the model. The video quality of unknown video sequences with a $1 \times m$ feature vector $\mathbf{x}_u$ can then be predicted as

$$\hat{y}_u = \hat{b}_0 + \mathbf{x}_u \hat{\mathbf{b}}. \tag{3}$$

For more information on PLS1 and PCA/PCR, we refer to [10] and to [11], respectively.

2.2. Trilinear Partial Least Squares Regression

The multidimensional extension of PLS1, multiway or N-way PLS, was introduced by Bro in [12]. It extends the principle behind PLS1 of maximizing the variance explained by the PCs in both sides of (1) to higher dimensional data. In particular, the trilinear partial least squares (Tri-PLS1) describes the partial least squares regression of a three-way $n \times m \times t$ data array $\mathbf{X}(:,:,:)$ onto an $n \times 1$ column (quality) vector $\mathbf{y}$. The main difference compared to PLS1 is that the principal components are now determined dependent on weights gained along both the m and t dimension, whereas in PLS1 the principal components are only dependent on the m dimension.

The iterative algorithm shown in Listing 2 describes how $\mathbf{X}(:,:,:)$ is decomposed in its PCs $\mathbf{w}^m$ and $\mathbf{w}^t$ along both feature dimensions. $\mathbf{Z}$ represents the matrix of all z_{mt}, with

$$z_{mt} = \sum_{n=1}^{N} y_n x_{nmt}. \tag{4}$$

The scores t_n corresponding to each sample n can then be written with the principal components as

$$t_n = \sum_{m=1}^{M} \sum_{t=1}^{T} x_{nmt} w_m^M w_t^T. \tag{5}$$

Algorithm 2: Trilinear PLS1

center $\mathbf{X}$ and $\mathbf{y}$
$\mathbf{y}_0 = \mathbf{y}$
$f = 1$

1 Calculate $\mathbf{Z}$

2 Determine $\mathbf{w}_f^m$ and $\mathbf{w}_f^t$ by SVD of Z

3 Calculate $\mathbf{t}_f$. $\mathbf{T} = [\mathbf{t}_1 \cdots \mathbf{t}_f]$

4 $\mathbf{b}_f = (\mathbf{T}^T \mathbf{T})^{-1} \mathbf{T} \mathbf{y}_0$

5 Each sample $\mathbf{X}_i$ is replaced with
$\mathbf{X}_i - t_i \mathbf{w}_f^m (\mathbf{w}_f^t)^T$ and $\mathbf{y} = \mathbf{y}_0 - \mathbf{T}\mathbf{b_f}$

6 Continue from 1 and let $f = f + 1$ until proper description of $\mathbf{y}_0$

Based on the extracted PCs and the scores, we can then obtain an estimation of a $t \times m$ regression matrix, $\hat{\mathbf{B}}$, for direct regression of a $1 \times m \times t$ feature slice of $\mathbf{X}(:,:,:)$, representing the features of a particular sequence over time on our quality vector $\mathbf{y}$.

Hence, the quality estimation (2) can now be written as

$$\hat{\mathbf{y}} = \mathbf{1}\hat{b}_0 + \mathbf{X}\hat{\mathbf{B}} + \mathbf{e}. \tag{6}$$

Unknown video sequences can be predicted similarly to (3), where the feature vector $\mathbf{x}_u$ is replaced by a corresponding feature slice $\mathbf{X}_u$. For a more detailed description of Tri-PLS1, we refer to [12, 13].

3. BITSTREAM BASED NO-REFERENCE METRIC

In order to compare PLS1 and Tri-PLS1, we will design a H.264/AVC bitstream based no-reference video quality metric with each method and then compare their prediction performance. The PLS1 based metric was already presented in [3].
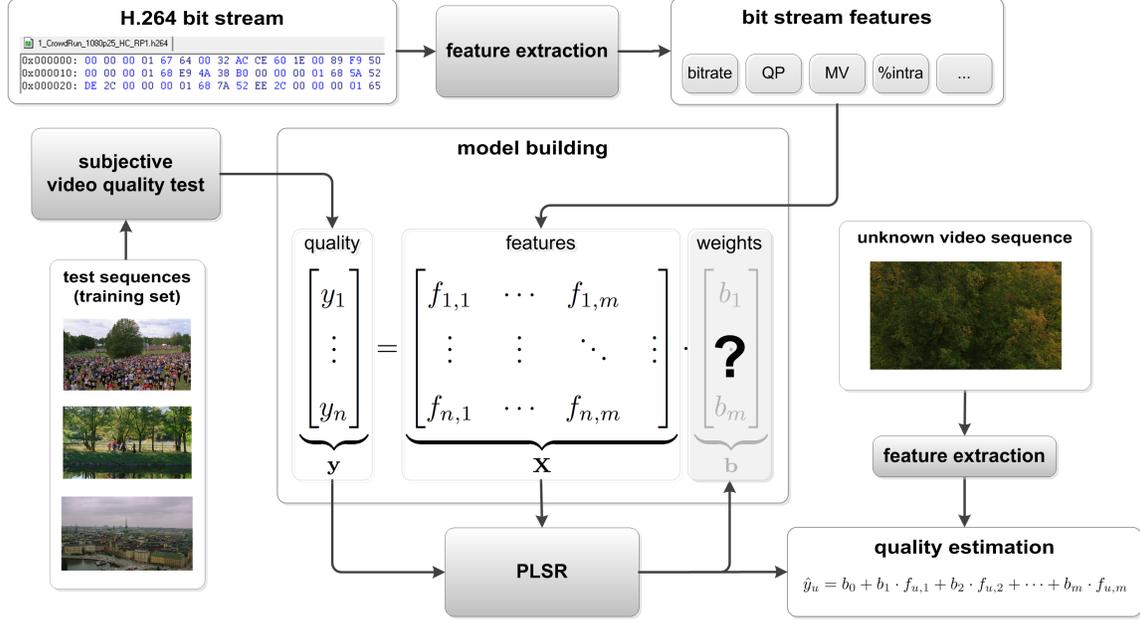
Fig. 2: Model building with bitstream features and PLSR

3.1. Feature Extraction

Firstly, we need to extract features from the H.264/AVC bitstream that describe the properties of the encoded video sequence. We assume in the following that the *byte stream* representing the Network Abstraction Layer (NAL) according to Annex B of the H.264/AVC standard is available and that any channel coding done for transmission has already been removed. Note that we do not have to further decode and reconstruct each frame: it is sufficient if we only reverse the entropy encoding of the bitstream, as we are not interested in the completely decoded frame, but rather in the properties of the bitstream. We then parse those NAL units (NALU) containing information about the coded frames, the so-called Video Coding Layer (VCL). Each VCL-NALU describes one slice of the current frame. A slice in turn is partitioned into multiple macroblocks, which again can be divided into submacroblocks. Hence, we parse three successive layers as shown in Fig. 3. Some of these features were also used by Rossholm and Lövström in [7]. For feature extraction, we used a modified decoder of the H.264/AVC reference software [14].

Before descending to the slice level, we extract the *profile*, *level* and *entropy encoding type* for the complete video sequence. Then, we extract the following features for each slice in the video sequence:

- slice type: I-, P- and B-slices (*%I,%P,%B*)
- bits per slice (*BPS*)
- average QP per slice (*QPA*)
- average, minimum and maximum motion vector length per slice (*MV, MVMin, MVMax*)
- average and maximum motion vector error per slice (*MVd, MVdMax*)

For the PLS1 based video quality metric, these features are then pooled temporally over all frames by calculating the average, median, standard deviation, minimum, maximum, 10% and 90% percentiles. We denote this for each feature f as f_{Avg}, f_{Med}, f_{SD},

$f_{Min/Max}$ and $f_{10/90}$. Additionally, we calculate the average difference qpd_{Avg} between initial and changed QP over all slices, but also the percentage of slices with constant QP over all macroblocks (*%qpd*). Furthermore, we determine the percentage of the different slice types, the different macroblock types and their subdivision over the whole video sequence:

- percentage of intra, inter and skip coded macroblocks (*%Intra, %Inter, %Skip*)
- percentage of intra macroblocks with 16×16, 8×8 and 4×4 subdivision (*%I16x16, %I8x8, %I4x4*)
- percentage of inter macroblocks with 8×8 and 4×4 subdivision (*%P8x8, %P4x4*)

All in all, we thus get 64 different features from each video sequence in the PLS1 case.

For the Tri-PLS1 based video quality metric, no temporal pooling is performed and we used all 19 features directly in the model building. Hence, we get an $m \times t$ feature matrix for each sequence. We did not consider features that did not vary during the sequence i.e. *profile*, *level* and *entropy encoding type*.

3.2. Sigmoid Correction

On the extremes of the voting scale in subjective testing, at very good or bad quality, the test results exhibit a nonlinear nature. Thus ratings do not reach the boundaries of the scale, but saturate earlier. Therefore, we correct the prediction values $\hat{y}$ slightly, using a fixed sigmoid nonlinear correction in order to emulate this behavior [1]. The sigmoid correction of $\hat{y}$ is given as

$$\hat{y}_S = 1.0/(1 + e^{(-(\hat{y}-0.5)/0.2)}). \tag{7}$$

This function is not adapted to the actual data, but is rather a fixed part of the quality metric. Hence, $\hat{y}_S$ represents the final prediction result of our video quality metric. The correction function is shown in Fig. 6.
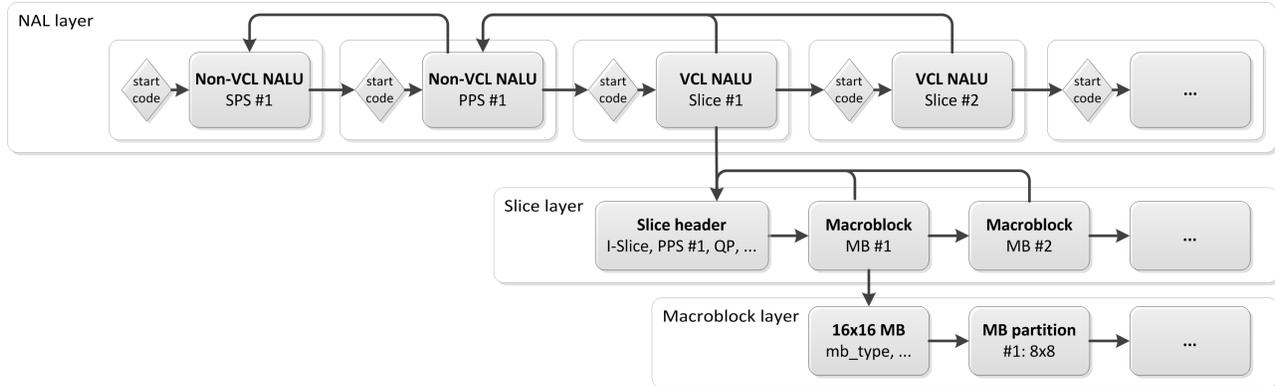
Fig. 3: H.264/AVC bitstream: overview over the different layers

3.3. Cross Validation

It is important to use separate data sets for training and validation of the designed metric. If we used the same data for training and validation, it would lead to overly optimistic prediction models as discussed in [1].

Therefore, we perform a cross validation and split the available data set into four different subsets and apply both data analysis methods on each of these subsets. Each subset consists of all data sets excluding one data set of the four video sequences introduced in the following section. Consequently, we compute four different models, allowing us to verify the results for each sequence using a model that did not include this particular sequence during the calibration.

4. SUBJECTIVE TESTING

For subjective testing, we encoded four different video sequences with the H.264/AVC reference encoder at multiple bitrates. We used two significantly different encoder settings, each representing the complexity of various devices and services. The first setting is chosen to simulate a low complexity (LC) H.264/AVC encoder: many tools that account for the high compression efficiency are disabled. In contrast to this, we also used a high complexity (HC) setting that aims at getting the maximum possible quality out of this coding technology. We used the H.264/AVC reference software [14] version 12.4. Selected encoding settings are listed in Table 1. We used the test sequences *CrowdRun*, *ParkJoy*, *InToTree* and *OldTownCross* from the SVT high definition multi format test set in the 1080p25 HDTV format, as shown in Fig. 7. We selected four rate points at bitrates from 5.4 Mbit/s to 30 Mbit/s. This resulted in a quality range from 'not acceptable' to 'perfect', corresponding to mean opinion scores (MOS) between 0.19 and 0.96 on a scale ranging from 0 to 1. In total, we have thus 32 different data points. The tests were performed in the video quality evaluation laboratory of the Institute for Data Processing at the Technische Universität München in a room compliant with recommendation ITU-R BT.500 [15]. For more details on the subjective testing and the used data set, we refer to [16].

5. MODEL BUILDING

The PLS1 regression and subsequent cross validation on the four subsets reveal that only 48 of the temporally pooled features are relevant for the model and thus $m = 48$, as the regression weights of the other 12 do not have any significant influence on the predicted

Table 1: Selected encoder settings

	LC	HC
Encoder	JM 12.4	
Profile&Level	Main, 4.0	High, 5.0
Slices per Frame	1	1
Reference Frames	2	5
R/D Optimization	Fast Mode	On
Search Range	32	128
B-Frames	2	5
Temporal Levels	2	4
Intra Period	500 ms	
8x8 Transform	Off	On

quality. In particular, we can exclude the feature *MVMin* completely. We determined an optimal number of 3 PCs to efficiently describe the variance in both **X** and **y** at the same time.

The video quality model with Tri-PLS1 showed after cross validation, that only 15 of the originally 19 extracted features play a significant role in the quality prediction. In particular, we could exclude the percentage of different slices (*%I,%P,%B*), *MVMin* and *BPS*. We used the complete sequences with all frames in the model building step. For the Tri-PLS1 model, we determined an optimal number of 4 PCs to describe the model sufficiently. The resulting regression matrix $\widehat{\mathbf{B}}$ is shown exemplary for the model built with the sequences *ParkJoy*, *InToTree* and *OldTownCross* in Fig. 5.

Fig. 5: Regression matrix $\widehat{\mathbf{B}}$ for *ParkJoy*, *InToTree* and *OldTown-Cross* model, showing frames 50 to 75
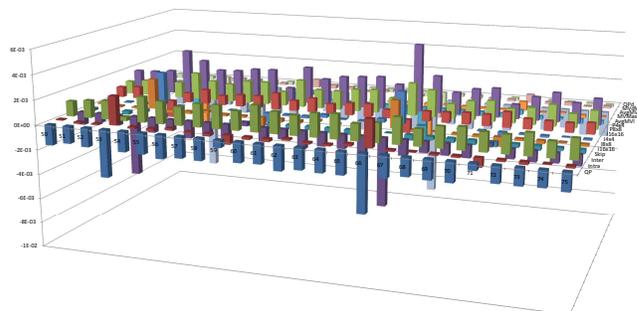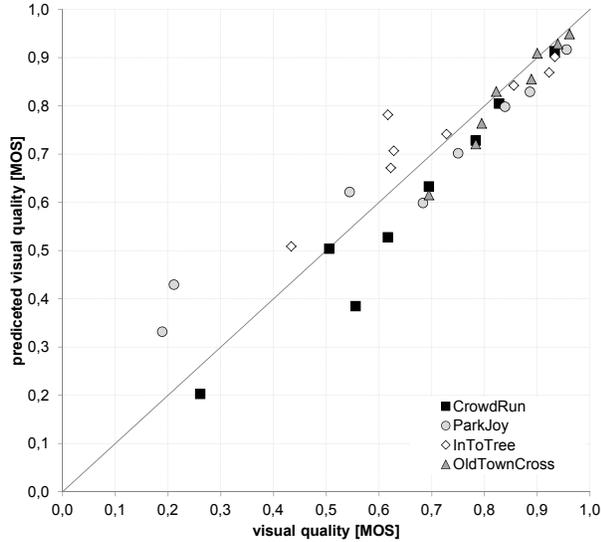
(a) PLS1
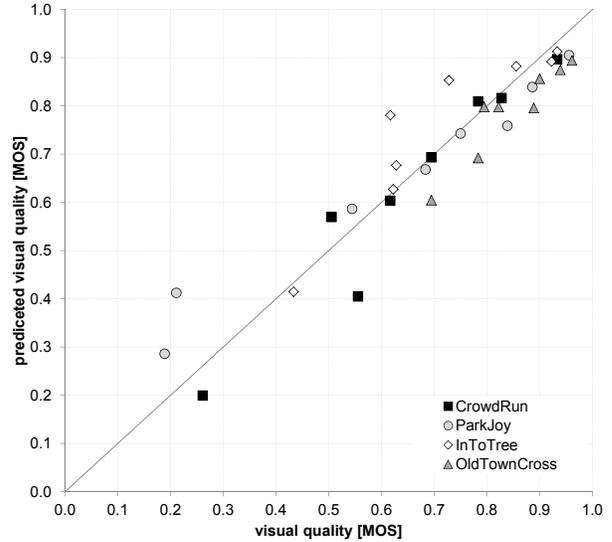
(b) Tri-PLS1

Fig. 4: Prediction results for both methods with sigmoid correction

For both PLS1 and Tri-PLS1, features were considered to be insignificant if they could be omitted from the model building without any overall influence on the prediction performance of the gained model. This indicates that the contribution of these feature is negligible for explaining the visual quality variance.

Note, that while the models designed with both methods are relatively independent of the content due to the cross validation, we only considered two encoder settings with two different coding structures. Hence, the models will not yield valid results for very different coding structures.

Table 2: Performance of the quality prediction

Metric	Pearson	Spearman	RMSE [a]
PLS1 based [b]	0.91	0.95	0.09
Tri-PLS1 based [b]	0.93	0.93	0.08
PLS1 based	0.93	0.95	0.08
Tri-PLS1 based	0.94	0.93	0.07
No-reference metric [1]	0.91	0.85	0.09
PSNR	0.72	0.69	0.15
SSIM [17]	0.85	0.82	0.12
VQM Annex D of [18]	0.84	0.78	0.11

[a] After first order fitting for all comparison metrics, no fitting for both no-reference metrics
[b] No sigmoid correction

6. PREDICTION PERFORMANCE

The prediction results of the metrics based on both methods are presented in Fig. 4 and Table 2. Besides the Pearson and Spearman rank order correlation coefficents, we also provide the root mean squared error (RMSE) between predicted and actual visual quality. For comparison, we included the results of our no-reference metric presented in [1], but also the results of two well-known full-reference video quality metrics: SSIM [17] and the VQM according to Annex D of ITU-T J.144 [18]. For the latter, the general model was used. SSIM was evaluated on all three channels of the YC_BC_R color space. While both metrics are general purpose metrics and therefore not as tuned to H.264/AVC artefacts as the proposed metric, they still provide a good baseline comparison to the state-of-the-art in video quality metrics.

The results show that both metrics outperform our previous no-reference metric in [1] slightly with respect to the Pearson correlation and the RMSE, but especially well with respect to the Spearman rank order correlation. This is not surprising, as we build our model only for H.264/AVC, compared to [1], where we also considered alternative, wavelet-based coding technologies.Moreover, our metrics outperformed all full-reference metrics. However, note in Fig. 4 that due to the lack of low quality data points in the training set, the prediction quality is worse at the lower end of the quality scale.

In comparing both our metrics, we notice that the metric designed with Tri-PLS1 slightly outperforms the PLS1 based metric with respect to the Pearson correlation and the RMSE, although PLS1 has a better prediction monotonicity as shown by the Spearman rank order correlation. In particular, the sigmoid correction seems to be less important for Tri-PLS1 than for PLS1: if we take a closer look at the individual data points before sigmoid correction, we can notice that while for the PLS1 models some predictions exceed the upper limit of the quality scale and thus the correction is needed to gain valid quality values, all predictions with the Tri-PLS1 models are within the valid range.

Also we can note in Fig. 4 that the prediction performance was especially increased for the sequences *CrowdRun* and *ParkJoy* with Tri-PLS1. In comparision, for the less demanding sequence *OldTownCross*, the prediction performance is slightly decreased for some rate points with Tri-PLS1. This can partly be explained by the lack of variance between different rate points in the training set for this sequence. For more details about the training set, we refer to [16].

7. CONCLUSION

We extended the design of video quality metrics with data analysis methods into the higher dimensional space for better exploitation of video's temporal nature. We designed two no-reference video quality metrics with both the bilinear partial least squares regression method, PLS1, and with its multidimensional extension trilinear PLS1.

Our results show that avoiding temporal pooling and the application of trilinear PLS1 increases the prediction accuracy of the designed metrics. Still, both metrics can be further improved in future work by either including a larger data set or by considering more than two different prediction structures and encoding settings.

The H.264/AVC bitstreams, the modified decoder for feature extraction, regression matrices for the different models and additional data is available at `www.ldv.ei.tum.de/videolab`.

8. REFERENCES

[1] C. Keimel, T. Oelbaum, and K. Diepold, "No-reference video quality evaluation for high-definition video," *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 1145–1148, Apr. 2009.

[2] ——, "Improving the prediction accuracy of video qualtiy metrics." *Acoustics, Speech and Signal Processing, 2010. ICASSP 2010. IEEE International Conference on*, pp. 2442–2445, Mar. 2010.

[3] C. Keimel, M. Klimpke, J. Habigt, and K. Diepold, "No-reference video quality metric for HDTV based on H.264/AVC bitstream features," in *Proceedings of the IEEE International Conference on Image Processing*, 2011, submitted.

[4] C. Keimel, M. Rothbucher, and K. Diepold, "Extending video quality metrics to the temporal dimension with 2D-PCR," S. P. Farnand and F. Gaykema, Eds., vol. 7867, no. 1. SPIE, 2011, p. 786713.

[5] 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, May 2007.

[6] M. Slanina, V. Ricny, and R. Forchheimer, "A novel metric for H.264/AVC no-reference quality assessment," in *EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, 2007, pp. 114 –117.

[7] A. Rossholm and B. Lövström, "A new video quality predictor based on decoder parameter extraction," in *International Conference on Signal Processing and Multimedia Applications*, 2008, pp. 285–290.

[8] S.-O. Lee, K.-S. Jung, and D.-G. Sim, "Real-time objective quality assessment based on coding parameters extracted from H.264/AVC bitstream," *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 1071 –1078, May 2010.

[9] O. Sugimoto, S. Naito, S. Sakazawa, and A. Koike, "Objective perceptual video quality measurement method based on hybrid no reference framework," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, 2009, pp. 2237 –2240.

[10] H. Martens and M. Martens, *Multivariate Analysis of Quality*. Wiley & Sons, 2001.

[11] I. Jolliffe, *Principal Component Analysis*. Springer, 2002.

[12] R. Bro, "Multiway calibration. multilinear PLS," *Journal of Chemometrics*, vol. 10, no. 1, pp. 47–61, 1996.

[13] ——, "Multi-way analysis in the food industry - models, algorithms, and applications," Ph.D. dissertation, University of Amsterdam, Netherlands, 1998.

[14] K. Sühring. (2007) H.264/AVC software coordination. [Online]. Available: http://iphome.hhi.de/suehring/tml/index.htm

[15] *ITU-R BT.500 Methodology for the Subjective Assessment of the Quality for Television Pictures*, ITU-R Std., Rev. 11, Jun. 2002.

[16] C. Keimel, J. Habigt, T. Habigt, M. Rothbucher, and K. Diepold, "Visual quality of current coding technologies at high definition IPTV bitrates," in *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, 2010, pp. 390 –393.

[17] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[18] "ITU-T J.144. objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference," Mar. 2004.

Fig. 6: Sigmoid correction

(a) CrowdRun   (b) ParkJoy

(c) InToTree   (d) OldTownCross
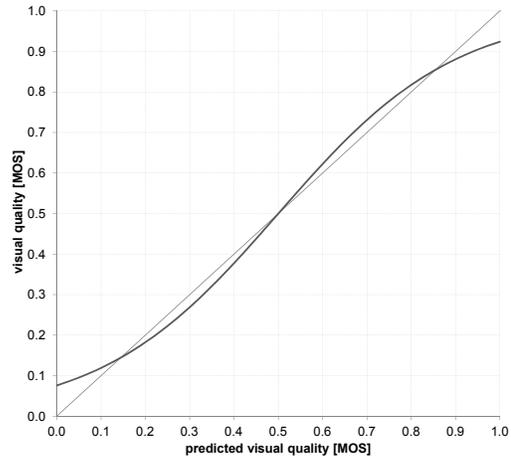
Fig. 7: Test sequences from the SVT high definition multi format test set