Video is a Cube: Multidimensional Analysis and Video Quality Metrics

Christian Keimel, Martin Rothbucher, Hao Shen and Klaus Diepold
Institute for Data Processing, Technische Universität München
Arcisstr. 21, 80333 München

***Abstract*—Quality of Experience is becoming increasingly important in signal processing applications. In taking inspiration from chemometrics, we provide an introduction to the design of video quality metrics by using data analysis methods, which are different from traditional approaches. These methods do not necessitate a complete understanding of the human visual system. We use multidimensional data analysis, an extension of well established data analysis techniques, allowing us to exploit higher dimensional data better. In the case of video quality metrics, it enables us to exploit the temporal properties of video more properly, the complete three dimensional structure of the video cube is taken into account in metrics' design. Starting with the well known principal component analysis and an introduction to the notation of multi-way arrays, we then present their multidimensional extensions, delivering better quality prediction results. Although we focus on video quality, the presented design principles can easily be adapted to other modalities and to even higher dimensional datasets as well.**

QUALITY OF EXPERIENCE (QoE) is a relatively new concept in signal processing that aims to describe how video, audio and multi-modal stimuli are perceived by human observers. In the field of video quality assessment, it is often of interest for researchers how the overall experience is influenced by different video coding technologies, transmission errors or general viewing conditions. The focus is no longer on measurable physical quantities, but rather on how the stimuli are subjectively experienced and whether they are perceived to be of acceptable quality from a subjective point of view.

QoE is in contrast to the well-established *Quality of Service (QoS)*. There, we measure the signal fidelity, i.e. how much a signal is degraded during processing by noise or other disturbances. This is usually done by comparing the distorted with the original signal, which then gives us a measure of the signal's quality. To understand the reason why QoS is not sufficient for capturing the subjective perception of quality, let us take a quick look at the most popular metric in signal processing to measure the QoS, the mean squared error (MSE). It is known that the MSE does not correlate very well with the human perception of quality, as we just determine the difference between pixel values in both images. The example in Fig. 1 illustrates this problem. Both images on the left have the same MSE with respect to the original image. Yet, we perceive the upper image distorted by coding artefacts to be of worse visual quality, than the lower image, where we just changed the contrast slightly. Further discussions of this problem can be found in [1].

I. How to Measure Quality of Experience

How then can we measure QoE? The most direct way is to conduct tests with human observers, who judge the visual quality of video material and provide thus information about the subjectively perceived quality. However, we face a problem in real-life: these tests are time consuming and quite expensive. The reason for this is that only a limited number of subjects can take part in a test at the same time, but also because a multitude of different test cases have to be considered. Apart from these more logistical problems, subjective tests are usually not suitable if the video quality is required to be monitored in real time.

To overcome this difficulty, video quality metrics are designed and used. The aim is to approximate the human quality perception as good as possible with objectively measurable properties of the videos. Obviously, there is no single measurable quantity that by itself can represent the perceived QoE. Nevertheless, we can determine some aspects which are expected or shown to have a relation to the perception of quality and use these to design an appropriate video quality metric.

II. Design of Video Quality Metrics - The Traditional Approach

In the traditional approach, the quality metrics aim to implement the spatial and temporal characteristics of the human visual system (HVS) as well as possible in our metric. Many aspects of the HVS are not sufficiently understood and therefore a comprehensive model of the HVS is hardly possible to be built. Nevertheless, at least parts of the HVS can be described sufficiently enough in order to utilize these properties in video quality metrics. In general, there are two different ways to exploit these properties according to Winkler [2]: either a *psychophysical* or an *engineering* approach.

The *psychophysical* approach relies primarily on a (partial) model of the HVS and tries to exploit known psychophysical effects, e.g. masking effects, adaption and contrast sensitivity. One advantage of this approach is that we are not limited to a specific coding technology or application scenario, as we implement an artificial observer with properties of the HVS. In Daly's visual differences predictor [3], for example, the adaption of the HVS at different light levels is taken into account, followed by an orientation dependent contrast sensitivity function and finally models of the HVS's different detection mechanism are applied. This full-reference predictor

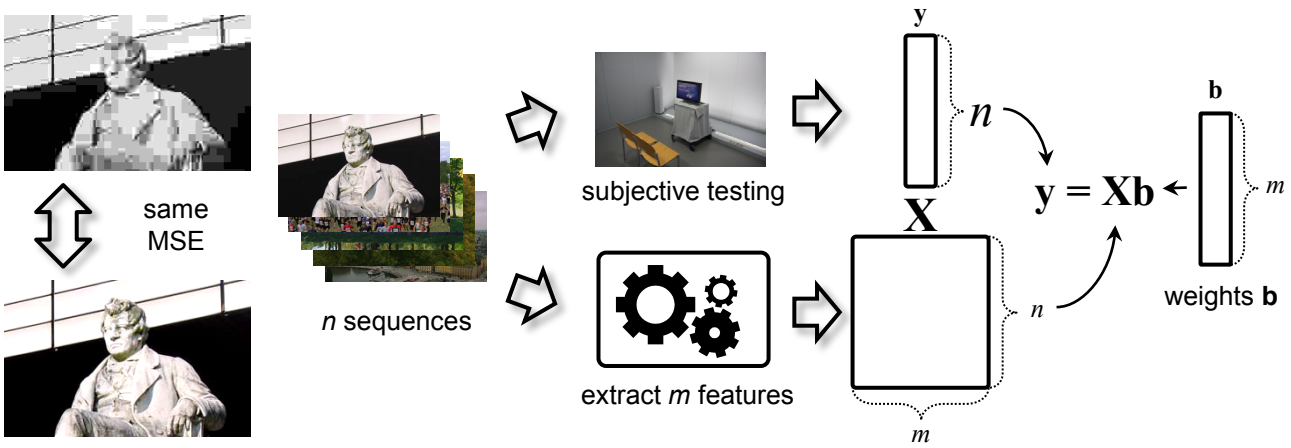

Fig. 1: Images with same MSE, but different visual quality (left) and how a model is built with data analysis: subjective testing and feature extraction for each video sequence (right)

then delivers a measure for the perceptual difference in the same areas of two images. Further representatives of this approach are Lubin's visual discrimination model [4], the Sarnoff Just Noticeable Difference (JND) [5] and Winkler's perceptual distortion metric [6].

In the *engineering* approach, the properties of the HVS are not implemented directly, but rather it determines features known to be correlated to the perceived visual quality. These features are then extracted and used in the video quality metric. As no in-depth understanding of the HVS is needed, this type of metric is commonly used in current research. In contrast to the psychophysical approach, however, we are limited to predefined coding technologies or application scenarios, as we do not construct an artificial observer, but rather derive the features from artifacts introduced by the processing of the videos. One example for such a feature is *blocking* as seen in Fig. 1. This feature is well known, as it is especially noticeable in highly compressed video. It is caused by block-based transforms such as the Discrete Cosine Transform (DCT) or integer transform in the current video encoding standards MPEG-2 and H.264/AVC. With this feature, we exploit the knowledge, that human perception is sensitive to edges, and assume therefore that artificial edges introduced by the encoding results in a degraded, perceived quality. Usually, more than one feature is extracted and these features are then combined into one value, by using assumptions about the HVS. Typical representatives of this approach are the widely used Structural Similarity (SSIM) index by Wang et al. [7], the video quality metric by Wolf and Pinson [8] and Watson's digital video quality metric [9]. Moreover we can distinguish between full-reference, reduced-reference and no-reference metrics, where we have either the undistorted original, some meta information about the undistorted original or only the distorted video available, respectively. We refer to [10] for further information on the HVS, and [11], [12] for an overview of current video quality metrics.

In general, the exploitation of more properties of the HVS or their corresponding features in a metric allows us to model the perception of quality better. However, since the HVS is not understood completely, and consequently, no explicit model of the HVS describing all its aspects is available in the community, it is not obvious how the features should be combined. But do we really need to know a-priori how to combine the features?

III. An Alternative Method: Data Analysis

Sometimes it is helpful to look at other disciplines. Video quality estimation is not the only application area in which we want to quantify something that is not directly accessible for measurement. Similar problems often occur in chemistry and related research areas. In food science, for example, researchers face a comparable problem: they want to quantify the taste of samples, but taste is not directly measurable. The classic example is about the determination of the perfect mixture for hot chocolate that tastes best. One can measure milk, sugar or cocoa content, but there is not an a-priori physical model that allows us to define the resulting taste. To solve this problem, a data-driven approach is applied, i.e. instead of making explicit assumptions of the overall system and relationship between the dependent variable, e.g. taste and the influencing variables e.g. milk, sugar and cocoa, the input and output variable are analyzed. In this way we obtain models purely via the analysis of the data.

In chemistry, this is known as *chemometrics* and has been applied successfully to many problems in this field for the last three decades. It provides a powerful tool to tackle the analysis and prediction of systems that are understood only to a limited degree. So far this method is not well known in the context of video quality assessment or even multimedia quality assessment in general. A good introduction into chemometrics can be found in [13].

By applying this multivariate data analysis to video quality, we now consider the HVS as a black box and therefore do not assume a complete understanding of it. The input corresponds to features we can measure and the output of the box to the perceived visual quality obtained in subjective tests. Firstly, we extract m features from an image or video frame I, resulting in a $1 \times m$ row vector $\mathbf{x}$. While this is similar to

the engineering approach described in the previous section, an important difference is that we do not make any assumption about the relationship between the features themselves, but also not about how they are combined into a quality value.

In general, we should not limit the number of selected features unnecessarily. Or to quote Martens and Martens [13], *Beware of wishful thinking!* As we do not have a complete understanding of the underlying system, it can be fatal if we exclude some features before conducting any analysis, because we consider them to be irrelevant. On the other hand, data that can be objectively extracted, like the features in our case, is usually cheap or in any case less expensive to generate than subjective data gained in tests. If some features are irrelevant to the quality, we will find out during the analysis. Of course it is only sensible to select features that have some verified or at least some suspected relation to the human perception of visual quality. For example, we could measure the room temperature, but it is highly unlikely that room temperature has any influence in our case.

For n different video sequences, we extract a corresponding feature vector $\mathbf{x}$ for each sequence and thus get an $n \times m$ matrix $\mathbf{X}$, where each row describes a different sequence or sample and each column describes a different feature as shown in Fig. 1. We generate a subjective quality value for each of the n sequences by subjective testing and get an $n \times 1$ column vector $\mathbf{y}$ that will represent our ground truth. Based on this dataset, a model can be generated to explain the subjectively perceived quality with objectively measurable features. Our aim is now to find an $m \times 1$ column vector $\mathbf{b}$ that relates the features in $\mathbf{X}$ to our ground truth in $\mathbf{y}$ or provides the weights for each feature to get the corresponding visual quality. This process is called *calibration* or *training* of the model, and the used sequences are the training set. We can use $\mathbf{b}$ to also predict the quality of new, previously unknown sequences. The benefit of using this approach is that we are able to combine totally different features into one metric without knowing their proportional contribution to the overall perception of quality beforehand.

IV. Classic and Well Known: Linear Regression

One classic approach to estimate the weight vector $\mathbf{b}$ is via a simple multiple linear regression model, i.e.

$$\mathbf{y} = \mathbf{X}\widehat{b} + \epsilon, \tag{1}$$

where ϵ is the *error term*. Without loss of generality, the data matrix $\mathbf{X}$ can be assumed to be centered, namely with zero means, and consequently the video quality values $\mathbf{y}$ are also centered.

Using a least squares estimation, we are given an estimation of $\mathbf{b}$ as

$$\widehat{\mathbf{b}} = (\mathbf{X}^\top\mathbf{X})^{+}\mathbf{X}^\top\mathbf{y}, \tag{2}$$

where Z^+ denotes the More-Penrose pseudo-inverse of matrix Z. We use the pseudo-inverse, as we can not assume that columns of $\mathbf{X}$ representing the different features are linearly independent and therefore $\mathbf{X}^\top\mathbf{X}$ can be rank deficient. For an unknown video sequence V_U and the corresponding feature vector $\mathbf{x}_u$, we are then able to predict its visual quality $\widehat{y}_u$ with

$$\widehat{y}_u = \mathbf{x}_u\widehat{\mathbf{b}}. \tag{3}$$

Yet, this simple approach has a drawback: we assume implicitly in the estimation process of the weights that all features are equally important. Clearly, this will not always be the case, as some features may have a larger variance than others.

V. An Improvement: Principal Component Regression

We can address the aforementioned issue by selecting the weights in the model, so that they take into account the influence of the individual features on the variance in the feature matrix $\mathbf{X}$. We are therefore looking for so-called *latent* variables, that are not directly represented by the measured features themselves, but rather by a hidden combination of them. In other words, we aim to reduce the dimensionality of our original feature space into a more compact representation, more fitting for our latent variables. One well known method is the Principal Component Analysis (PCA), which extracts the latent variables as the Principal Components (PCs). The variance of the PCs is expected to preserve the variance of the original data. We then perform a regression on some of these PCs leading to the principal component regression (PCR). As PCA is a well known method, we just briefly recap some basics.

Let $\mathbf{X}$ be a (centered) data matrix, we define $r = \min\{n, m\}$ and using a singular value decomposition (SVD) we get the following factorization:

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{P}^\top, \tag{4}$$

where $\mathbf{U}$ is an $n \times r$ matrix with r orthonormal columns, $\mathbf{P}$ is an $m \times r$ matrix with r orthonormal columns and $\mathbf{D}$ is a $r \times r$ diagonal matrix. $\mathbf{P}$ is called the *loadings* matrix and its columns $\mathbf{p}_1, \ldots, \mathbf{p}_r$ are called loadings. They represent the eigenvectors of $\mathbf{X}^\top\mathbf{X}$. Furthermore we define the *scores* Matrix

$$\mathbf{T} = \mathbf{U}\mathbf{D} = \mathbf{X}\mathbf{P}. \tag{5}$$

The basic idea behind PCR is to approximate $\mathbf{X}$ by only using the first g columns of $\mathbf{T}$ and $\mathbf{P}$, representing the g largest eigenvalues of $\mathbf{X}^\top\mathbf{X}$ and also the first g PCs. We hereby assume that the combination of the largest g eigenvalues describe the variance in our data matrix $\mathbf{X}$ sufficiently and that we can therefore discard the smaller eigenvalues. If g is smaller than r, the model can be built with a reduced rank. Usually we aim to explain at least 80-90% of the variance in $\mathbf{X}$. But also other selection criteria are possible.

Our regression model with the first g PCs can thus be written as

$$\mathbf{y} = \mathbf{T}_g\mathbf{c}, \tag{6}$$

where $\mathbf{T}_g$ represents a matrix with the first g columns of $\mathbf{T}$ and $\mathbf{c}$ the (unknown) weight vector. Once again, we perform a multiple linear regression. We estimate $\mathbf{c}$ with the least squares method as

$$\widehat{\mathbf{c}} = (\mathbf{T}_g^\top\mathbf{T}_g)^{-1}\mathbf{T}_g^\top\mathbf{y}. \tag{7}$$

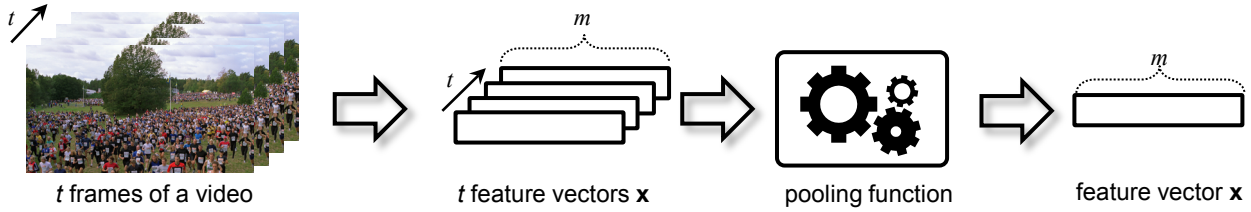

Fig. 2: Temporal pooling

In the end, we are interested in the weights, so that we can directly calculate the visual quality. Therefore we determine the estimated weight vector $\widehat{\mathbf{b}}$ as

$$\widehat{\mathbf{b}} = \mathbf{P}_g \widehat{\mathbf{c}}, \tag{8}$$

with $\mathbf{P}_g$ representing the matrix with the first g columns of $\mathbf{P}$. We can predict the visual quality for an unknown video sequence V_U and the corresponding feature vector $\mathbf{x}_u$ with (3).

PCA was firstly used in the design of video quality metrics by Miyahara in [14]. We refer to [15] for further information on PCA and PCR. A more sophisticated method often used in chemometrics is the partial least squares regression (PLSR). This method also takes the variance in the subjective quality vector $\mathbf{y}$ into account as well as the variance in the feature matrix $\mathbf{X}$. PLSR has been used in the design of video quality metrics in e.g. [16]. Further information on PLSR itself can be found in [13] and [17].

VI. Video is a Cube

The temporal dimension is the main difference between still images and video. In the previous section we assumed that we extract the feature vector only from one image or one video frame, which is a two dimensional matrix. In other words, video was considered to be just a simple extension of still images. This is not a unique omission only in this article so far, but the temporal dimension is quite often neglected in many contributions in the field of video quality metrics. The additional dimension is usually managed by temporal pooling. Either the features themselves are temporally pooled into one feature value for the whole video sequence or the metric is applied to each frame of the video separately and then the metric's values are pooled temporally over all frames to gain one value, as illustrated in Fig. 2.

Pooling is mostly done by averaging, but also other simple statistical functions are employed such as standard deviation, 10/90% percentiles, median or minimum/maximum. Even if a metric considers not only the current frame, but also preceding or succeeding frames, e.g. with a 3D filter [18] or spatio-temporal tubes [19], the overall pooling is still done with one of the above functions. But this arbitrary pooling, especially averaging, obscures the influence of temporal distortions on the human perception of quality, as intrinsic dependencies and structures in the temporal dimension are disregarded. The importance of video features' temporal properties in the design of video quality metrics was recently shown in [20].

Omitting the temporal pooling step and introducing the additional temporal dimension directly in the design of the video quality metrics can improve the prediction performance. We propose therefore to consider video in its natural three dimensional structure as a *video cube*. Extending the data analysis approach, we add an additional dimension to our dataset and thus arrive at multidimensional data analysis, an extension of the two dimensional data analysis. In doing so, we gain a better understanding of the video's properties and will thus be able to interpret the extracted features better. We no longer employ an a-priori temporal pooling step, but use the whole video cube to generate the prediction model for the visual quality and thus consider the temporal dimension of video more appropriately.

VII. Tensor Notation

Before moving on to the multidimensional data analysis, we shortly introduce the notation for handling *multi-way arrays* or *tensors*.

In general, our video cube can be presented as a three-way $u \times v \times t$ array $\mathbf{V}(:,:,:)$, where the u and v are the frame size, and t is the number of frames. Similarly, we can extend the two dimensional feature matrix $\mathbf{X}$ into the temporal dimension as a $n \times m \times t$ three-way array or *feature cube*. Both are shown in Fig. 3. In this work, we denote $\mathbf{X}(i,j,k)$ as the (i,j,k)-th entry of $\mathbf{X}$, $\mathbf{X}(i,j,:)$ as the vector with a fixed pair of (i,j) of $\mathbf{X}$, referred to as *tensor fiber*, and $\mathbf{X}(i,:,:)$ the matrix of $\mathbf{X}$ with a fixed index i, referred to as *tensor slice*. The different fibers and slices are shown in Fig. 4. For more information about tensors and multi-way arrays, see [21] and for multi-way data analysis refer to [22], [23].

VIII. Unfolding Time

The easiest way to apply conventional data analysis methods for analyzing tensor data, is to represent tensors as matrices. It transforms the elements of a tensor or multi-way array into

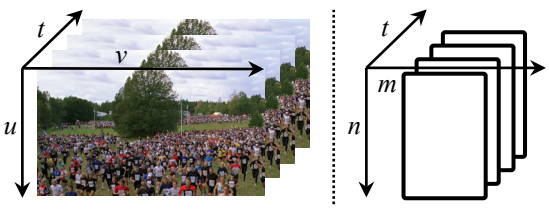

Fig. 3: Video cube and feature cube

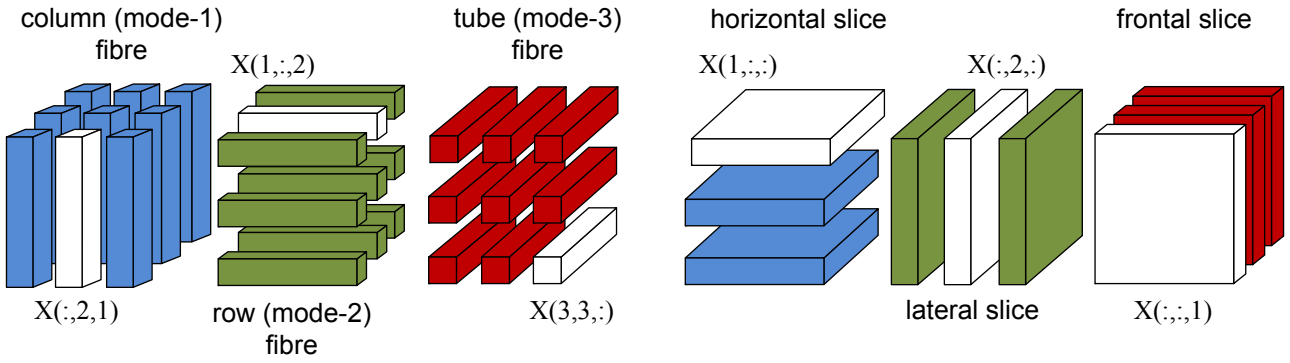

Fig. 4: Tensor notation: fibre (left) and slice (right)

entries of a matrix. Such a process is known as *unfolding*, *matricization*, or *flattening*.

In our setting, we are interested in the temporal dimension and therefore perform the mode-1 unfolding of our three-way array $\mathbf{X}(i,j,k)$. Thus we obtain a new $n \times (m \cdot t)$ matrix $\mathbf{X}_{unfold}$, whose columns are arranged mode-1 fibers of $\mathbf{X}(i,j,k)$. For simplicity, we assume that the temporal order is maintained in $\mathbf{X}_{unfold}$. The structure of this new matrix is shown in Fig. 5. We then perform a PCR on this matrix as described previously and obtain a model of the visual quality. Finally, we can predict the visual quality of an unknown video

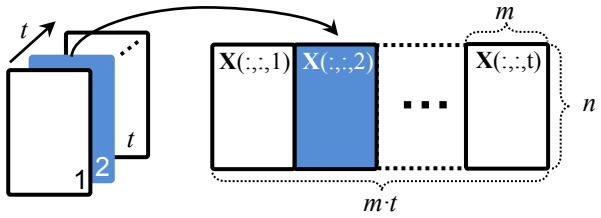

Fig. 5: Unfolding of the feature cube

sequence V_U with its feature vector $\mathbf{x}_u$ by using (3).

Note, that $\mathbf{x}_u$ is now of the dimension $1 \times (m \cdot t)$. One disadvantage during the model building step with PCR is that the SVD must be performed on a rather large matrix. Depending on the frames in the video sequence, the time needed for model building can increase by a factor of 10^3 or higher. But more importantly, we still lose some information about the variance by unfolding and thus destroying the temporal structure.

IX. 2D Principal Component Regression

Instead of unfolding we can include the temporal dimension directly in the data analysis via performing a *multidimensional data analysis*. We use the two-dimensional extension of the PCA, the (2D-PCA), recently proposed by Yang et al. [24], in combination with a least squares regression as 2D-PCR. For a video sequence with t frames, we can carve the $n \times m \times t$ feature cube into t slices, where each slice represents one frame. Without loss of generality, we can compute the covariance or *scatter* matrix as

$$\mathbf{X}_{Sct} = \frac{1}{t}\sum_{i=1}^{t} \mathbf{X}(:,:,i)^\top \mathbf{X}(:,:,i), \tag{9}$$

where, by abusing the notation, $\mathbf{X}(:,:,i)$ denotes the centered data matrix. It describes therefore the average covariance over the temporal dimension t. Then we perform the SVD performed on $\mathbf{X}_{Sct}$ to extract the PCs, similar to the previously described one dimensional PCR in (4).

Instead of a scores matrix $\mathbf{T}$, we now have a three-way $n \times g \times t$ scores array $\mathbf{T}(:,:,:)$, with each slice defined as

$$\mathbf{T}(:,:,i) = \mathbf{X}(:,:,i)\mathbf{P}. \tag{10}$$

Similar to (7), we then estimate a $g \times 1 \times t$ prediction weight for each slice with the first g principal components as

$$\widehat{\mathbf{C}}(:,:,i) = \left(\mathbf{T}_g(:,:,i)^\top \mathbf{T}_g(:,:,i)\right)^+ \mathbf{T}_g(:,:,i)^\top \mathbf{y}(i), \tag{11}$$

before expressing the weights in our original feature space with a $m \times 1 \times t$ three-way array

$$\widehat{\mathbf{B}}(:,:,i) = \mathbf{P}_g \widehat{\mathbf{C}}(:,:,\mathbf{i}), \tag{12}$$

comparable to (8) for the one dimensional PCR. Note, that the weights are now represented by a (rotated) matrix.

A quality prediction for the i-th slice can then be performed in the same manor as in (3), i.e.

$$\widehat{\mathbf{y}}_u(i) = \mathbf{X}_u(:,:,i)\widehat{\mathbf{B}}(:,:,i), \tag{13}$$

where $\mathbf{X}_u$ represents a $1 \times m \times t$ feature matrix for one sequence and $\widehat{\mathbf{y}}_u(i)$ the $1 \times t$ predicted quality vector. We can now use this quality prediction individually for each slice or generate one quality value for the whole video sequence by pooling. 2D-PCR has been used so far for video quality metrics in [25].

X. Cross Validation

In general data analysis methods require a training phase or a training set. One important aspect is to employ a separate data set for the validation of the designed metric. Using the same data set for training and validation will usually give us misleading results. Not surprisingly, our metric performs excellently with its training data. For unknown video sequences, on the other hand, the prediction quality could be very bad. But

Fig. 6: Cross validation: splitting up the data set in training and validation set (left), building different models with each combination (center) and quality prediction of a metric (right)

as mentioned previously, the data in video quality metrics is usually expensive to generate as we have to conduct subjective tests. Hence we can not really afford to use only a sub-set of all available data for the model building, as the more training data we have, the better the prediction abilities of our metric will be.

This problem can be partially avoided by performing a cross validation, e.g. *leave-one-out*. This allows us to use all available data for training, but also to use the same data set for the validation of the metric. Assuming we have i video sequences with different content, then we use $i-1$ video sequences for training and the left out sequence for validation. All in all, we eventually get i models. This is illustrated in Fig. 6. The general model can then be obtained by different methods e.g. averaging the weights over all models or selecting the model with the best prediction performance. For more information on cross validation in general we refer to [13] and [26].

XI. Using Data Analysis: An Example Metric

But do more dimensions really help us in designing better video quality metrics? In order to compare the approaches to data analysis we presented in this work, we therefore design as an example a simple metric for estimating the visual quality of coded video with each method in this section.

The cheapest objective data available for an encoded video can be found directly in its bitstream. Even tough we do not know a-priori which of the bitstream's properties are more, and which are less important, we can safely assume that they are related in some way to the perceived visual quality. How they are related, will be determined by data analysis. In this example, we use videos encoded with the popular H.264/AVC standard, currently used in many applications from high definition HDTV to internet based IPTV. For each frame, we extract 16 different features describing the partitioning into different block sizes and types, the properties of the motion vectors and lastly the quantization, similar to the metric proposed in [27]. Each frame is thus represented as 1×16 feature vector $\mathbf{x}$. Note, that no further preprocessing of the bitstream features was done. Alternatively, one can also extract features independent of the used coding technology, e.g. blocking or blurring, as described in [16].

Certainly, we also need subjective quality values for these encoded videos as ground truth in order to perform the data analysis. Different methodologies as well as the requirements on the test set-up and equipment for obtaining this data are described in international standards e.g. ITU-R BT.500 or ITU-T P.910. Another possibility is to use existing, publicly available datasets, containing both the encoded videos and the visual quality values. One advantage of using such datasets is that different metrics can be compared more easily.

For this example, we will use a dataset provided by IT-IST [28]. It consists of eleven videos in CIF resolution (352×288) and a frame rate of 30 frames per second as shown in Fig. 7. They cover a wide range of different content types, at bitrates from 64 kBit/s to 2.000 kBit/s, providing a wide visual quality range with in total $n = 52$ data points, leading to a 52×1 quality vector $\mathbf{y}$. According to [28], the test was conducted using to the DCR double stimulus method described in ITU-T P.910. For each data point, the test subjects were shown the undistorted original video, followed by the distorted encoded video, and then asked to assess the impairment of the coded video with respect to the original on a discrete five point mean opinion score (MOS) scale from 1, *very annoying*, to 5, *imperceptible*. For more information on H.264/AVC in general we refer to [29], and for the H.264/AVC feature extraction to [27]. A comprehensive list of publicly available datasets is provided at [30].

XII. More Dimensions are Really Better

Finally, we compare the four video quality metrics, each designed with one of the presented methods. By using a cross validation approach, we design eleven different models for each method. Each model is trained using ten video sequences and the left out sequence is then used for validation of the model built with the training set. Hence, we can measure the prediction performance of the models for unknown video sequences.

The performance of the different models is compared by calculating the Pearson correlation and the Spearman rank order correlation between the subjective visual quality and the

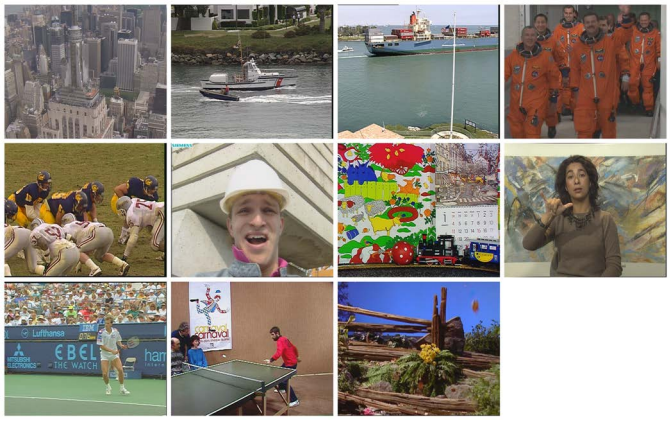

Fig. 7: Test videos from top to bottom, left to right: *City, Coastguard, Container, Crew, Football, Foreman, Mobile, Silent, Stephan, Table Tennis* and *Tempete*.

quality predictions. The Pearson correlation gives an indication about the prediction accuracy of the model and the Spearman rank order correlation gives an indication how much the ranking between the sequences changes between the predicted and subjective quality. Additionally, we determine the root mean squared error (RMSE) between prediction and ground truth, but also the percentage of predictions that fall outside the used quality scale from 1 to 5.

By comparing the results in Fig. 8 and Table I, we can see that a better inclusion of the temporal dimension in the model building helps to improve the prediction quality. Note, that this improvement was achieved very easily, as we did nothing else, but just changing the data analysis method. In each step we exploit the variation in our data better. Firstly just within our temporally pooled features with the step from multiple linear regression to PCR, then by the step in the third dimension with unfolding and 2D-PCR.

XIII. Summary

In this work, we provide an introduction into the world of data analysis and especially the benefits of multidimensional data analysis in the design of video quality metrics. We have seen in our example, that even with a very basic metric, by using multidimensional data analysis, we can increase the performance of predicting the *Quality of Experience* significantly. Although the scope of this introduction covered only the quality of video, the proposed methods can obviously be extended to more dimensions and/or other areas of application. It is interesting to note, that the dimensions need not be necessarily spatial or temporal, but also may represent different modalities or perhaps even a further segmentation of the existing feature spaces.

References

[1] Z. Wang and A. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98 –117, Jan. 2009.

[2] S. Winkler, *Digital Video Image Quality and Perceptual Coding*. CRC Press, 2006, ch. Perceptual Video Quality Metrics - A Review, pp. 155–179.

[3] S. J. Daly, *Digital Images and Human Vision*. MIT Press, 1993, ch. The visible differences predictor: An algorithm for the assessment of image fidelity, pp. 179–206.

[4] J. Lubin, *Vision Models for Target Detection and Recognition*. World Scientific Publishing, 1995, ch. A visual discrimination model for iamging system design and evaluation, pp. 245–283.

[5] J. Lubin and D. Fibush, *Sarnoff JND vision model*, T1A1.5 Working Group, ANSI T1 Standards Committee Std., 1997.

[6] S. Winkler, *Digital Video Quality - Vision Models and Metrics*. Wiley & Sons, 2005.

[7] 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.

[8] S. Wolf and M. H. Pinson, "Spatial-temporal distortion metric for in-service quality monitoring of any digital video system," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Multimedia Systems and Applications II*, vol. 3845, Nov. 1999, pp. 266–277.

[9] A. B. Watson, "Toward a perceptual video-quality metric," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Human Vision and Electronic Imaging III*, vol. 3299, Jan. 1998, pp. 139–147.

[10] B. A. Wandell, *Foundations of Vision*. Sinauer Associates, 1996.

[11] H. R. Wu and K. R. Rao, Eds., *Digital Video Image Quality and Perceptual Coding*. CRC Press, 2006.

[12] Z. Wang and A. C. Bovik., *Modern Image Quality Assessment*, ser. Synthesis Lectures on Image, Video, and Multimedia Processing. Morgan & Claypool Publishers, 2006.

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

[14] M. Miyahara, "Quality assessments for visual service," *IEEE Communications Magazine*, vol. 26, no. 10, pp. 51 –60, 81, Oct. 1988.

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

[16] T. Oelbaum, C. Keimel, and K. Diepold, "Rule-based no-reference video quality evaluation using additionally coded videos," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 294 –303, April 2009.

[17] F. Westad, K. Diepold, and H. Martens, "QR-PLSR: Reduced-rank regression for high-speed hardware implementation," *Journal of Chemometrics*, vol. 10, pp. 439–451, 1996.

[18] K. Seshadrinathan and A. Bovik, "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Transactions on Image Processing*, vol. 19, no. 2, pp. 335 –350, Feb. 2010.

[19] A. Ninassi, O. Le Meur, P. Le Callet, and D. Barba, "Considering temporal variations of spatial visual distortions in video quality assessment," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 253 –265, Apr. 2009.

[20] C. Keimel, T. Oelbaum, and K. Diepold, "Improving the prediction accuracy of video qualtiy metrics." *IEEE International Conference on Acoustics, Speech and Signal Processing, 2010. ICASSP 2010.*, pp. 2442–2445, Mar. 2010.

[21] A. Cichocki, R. Zdunek, A. H. Phan, and S.-I. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley & Sons, 2009.

[22] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis: Applications in the Chemical Sciences*. Wiley & Sons, 2004.

[23] P. M. Kroonenberg, *Applied Multiway Data Analysis*. Wiley & Sons, 2008.

[24] J. Yang, D. Zhang, A. Frangi, and J. Yang, "Two-dimensional pca: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131 –137, Jan. 2004.

[25] C. Keimel, M. Rothbucher, and K. Diepold, "Extending video quality metrics to the temporal dimension with 2D-PCR," in *Image Quality and System Performance VIII*, S. P. Farnand and F. Gaykema, Eds., vol. 7867. SPIE, Jan. 2011.

[26] E. Anderssen, K. Dyrstad, F. Westad, and H. Martens, "Reducing over-optimism in variable selection by cross-model validation," *Chemometrics and Intelligent Laboratory Systems*, vol. 84, no. 1-2, pp. 69–74, 2006.

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

[28] T. Brandão and M. P. Queluz, "No-reference quality assessment of H.264/AVC encoded video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1437–1447, Nov. 2010.

[29] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

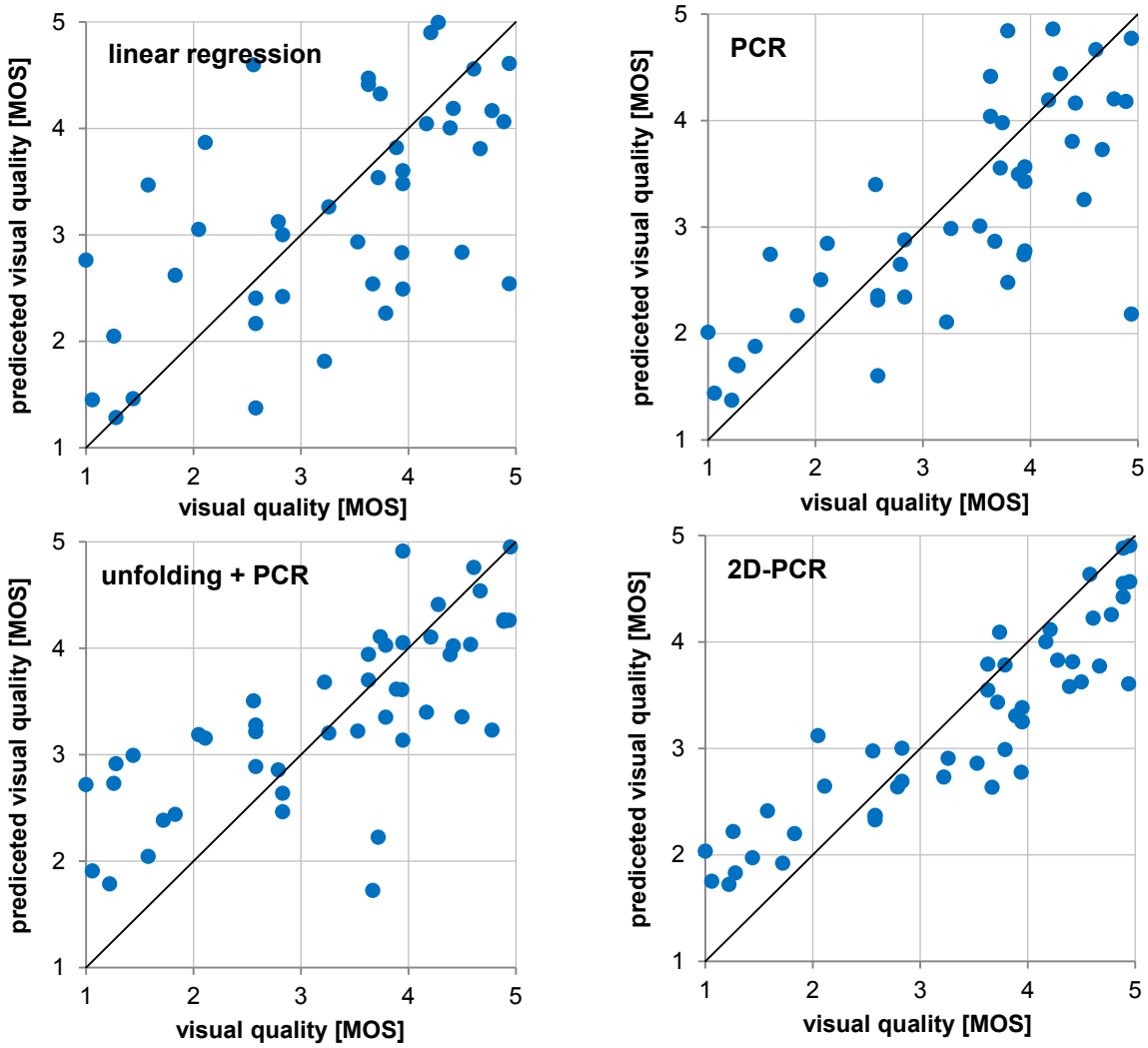

Fig. 8: Comparison of the presented methods: subjective quality vs. predicated quality on a mean opinion score (MOS) scale from 1 to 5, worst to best quality.

	Pearson correlation	Spearman correlation	RMSE	Outside scale
linear regression	0.72	0.72	1.04	15%
PCR	0.80	0.82	0.81	13%
unfolding + PCR	0.75	0.83	0.82	10%
2D-PCR	0.89	0.94	0.59	6%

TABLE I: Performance measurements: Pearson correlation, Spearman rank order correlation and RMSE. Additionally, the ratio of how many quality predictions are outside of the given scale.

[30] S. Winkler. (2011, Jul.) Image and video quality resources. [Online]. Available: http://stefan.winkler.net/resources.html