Technische Universität München
Lehrstuhl für Datenverarbeitung

Design of Video Quality Metrics with Multi-way Data Analysis

Christian Keimel

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzende(r): Univ.-Prof. Dr.-Ing. Sandra Hirche

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Klaus Diepold
2. Prof. Dr. Ir. Peter Schelkens, Vrije Universiteit Brussel, Brüssel, Belgien (schriftliche Beurteilung)
2. Univ.-Prof. Dr.-Ing. habil. Dirk Wollherr (mündliche Prüfung)

Die Dissertation wurde am 20. August 2013 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 24. März 2014 angenommen.

Christian Keimel. *Design of Video Quality Metrics with Multi-way Data Analysis.* Dissertation, Technische Universität München, Munich, Germany, 2014.

Institute for Data Processing, Technische Universität München, 80290 München, Germany, http://www.ldv.ei.tum.de.

Abstract

Video quality metrics determine the visual quality of distorted video sequences by using prediction models based on objectively measurable features and are therefore an alternative to the time-consuming and costly subjective video quality assessment. In the conventional design approach to video quality metrics, however, the temporal nature of video is often considered only inadequately due to the use of temporal pooling in the prediction process. Moreover, this approach also often requires knowledge about the human visual system that is not readily or only partly available. In this thesis, I therefore propose a data driven design methodology using multi-way data analysis for the design of video quality metrics. This data driven design approach not only requires no detailed knowledge of the human visual system, but also allows for a proper consideration of the temporal nature of video by using a three-way prediction model, corresponding to the three-way structure of video. Using two simple example metrics, I demonstrate that this purely data driven approach not only outperforms video quality metrics in the state-of-the-art that are often highly optimised towards specific properties of the human visual system, but also that multi-way data analysis methods outperform the combination of two-way data analysis methods and temporal pooling.

Contents

1. Introduction

Visual quality is something that human observers can intuitively judge by just looking at an image or video and using the two distorted images above as an example, most people would prefer the left image. Yet this simple task for humans, is not so simple for algorithms. Using the mean squared error, for example, as one of the most common approaches in engineering to assess the *goodness* of something with respect to a reference, both images should be considered as equally good, as the mean squared error of both images is the same. But clearly, this is not the case. This problem of assessing visual quality algorithmically has led to the research area of visual quality metrics that aim at providing algorithms allowing us to gain a measure of the visual quality as it would be perceived by human observers. Visual quality metrics can be divided into two fundamental groups: *image quality metrics* and *video quality metrics*. Image quality metrics aim at predicting the visual quality of one single image, in contrast to video quality metrics that aim at predicting the visual quality of a series of temporally consecutive and related images, represented by a video sequence and its constituting frames.

Research so far resulted in a multitude of image quality metrics using many different conceptual approaches to the visual quality assessment task, but far fewer video quality metrics. Due to this scarcity of specialised video quality metrics, image quality metrics are therefore also used for video quality assessment on a frame-by-frame basis by interpreting each frame as a separate image. Unfortunately, this practice often leads to a blurring between these two different groups of visual quality metrics in everyday use, but one should be aware of the fundamental difference of still images and video.

1.1. Motivation and problem statement

Visual quality metrics should fulfil three basic requirements: they should be able to provide a visual quality estimation similar to human observers, be properly validated and lastly should be applicable regardless of the availability of an undistorted reference in order to be usable in real-life applications. Video quality metrics should additionally also take the temporal nature of video and the resulting variation of visual quality over time properly into account and thus avoid any temporal pooling. Considering the video quality metrics proposed so far in the state-of-the-art, however, these four requirements are often only met partially:

Temporal nature of video Many video quality metrics account for the temporal nature of video by using pooling approaches as the Minkowski summation [21, 58, 183, 343, 349, 353, 354] or percentile pooling [27, 116, 190], while others include frames in a small interval around the current frame into the prediction [21, 183, 282] or use the differences to the preceding frame in the prediction [78, 212, 224, 227, 228, 232, 277, 341, 370, 373]. Only a small subset of all metrics consider the temporal dimension of video without any significant temporal pooling or at least consider a relatively large interval of a few seconds [11, 135, 172, 203, 220, 332].

Validation Validation of visual quality metrics is mostly limited to a single data set and only a small number of quality metrics are validated with multiple datasets [218, 232, 279, 302, 340, 373], but none of the video quality metrics are.

Reference availability Only a limited number of video quality metrics do not require the undistorted reference [23–25, 57, 58, 70, 135, 178, 224, 313, 314, 370]. Compared to the overall number of technology-agnostic video quality metrics in the-state-of-the-art, only a small subset fulfils this requirement [57, 58, 70, 135, 224, 370], in contrast to coding technology-specific metrics that in the majority fulfil this requirement [23–25, 178, 313, 314].

Prediction performance Using a Pearson correlation coefficient above 0.9 between the visual quality perceived by human observers and the prediction provided by the video quality metrics as a criterion for the metrics' prediction performance, only a small number of technology-agnostic metrics reach this goal [27, 70, 172, 190, 232, 277, 341], in contrast to technology-specific metrics that due to the focus on a specific technology are generally able to achieve the required prediction performance [23–25, 313, 314].

Although some video quality metrics in the state-of-the-art fulfil a subset of the four requirements, no metric fulfils all requirements in particular with respect to the proper consideration of the temporal nature of video.

1.2. Contribution of this thesis

In this thesis, I therefore propose a data driven design methodology using multi-way data analysis methods for the design of video quality metrics in order to address these shortcomings in the state-of-the-art, especially with respect to the inadequate consideration of video's temporal domain so far. Moreover, I demonstrate that this approach in combination with simple features and proper validation allows us to easily design metrics that are able to fulfil all four requirements outlined in the previous section.

Data driven design methodology Unlike the concept behind most video quality metrics in the state-of-the-art, the design methodology proposed in this thesis is purely *data driven*. It neither assumes nor needs a model of the human visual systems but utilises only objectively measurable properties of the video sequences, the so-called *features*. These features are then used in combination with visual quality ratings gained from human observers to built prediction models for the visual quality in a training phase. Hence, no a-priori relationship between certain features and visual quality is assumed, but the relationships between features and visual quality emerge during the training. The resulting models are then validated with independent, unknown sequences in order to assess the prediction accuracy. This data driven approach is widely used in *chemometrics*, a research area that often faces similar challenges to visual quality assessment in building prediction models for estimating the response of human perception to olfactory or gustatory stimuli.

In this thesis, I provide an detailed overview how a data driven design paradigm can be used in the design of video quality metrics. The major advantage of this approach is the separation of the feature extraction and model building process, allowing for the use of any features, using any model building method. Hence, even if only a partial knowledge of the influence of the video sequences' properties on the visual quality as perceived by human observers is available or some influence is maybe only suspected, the resulting features representing these properties can still be used in building a model, as now the overall combination of these features in a prediction model is independent from a a-priori model describing the exact relationship between the features.

Data driven design with multi-way data analysis Prediction models in the data driven design can be built with many different methods, but in this thesis I focus on the data analysis approach as commonly used in chemometrics. Data analysis methods have already previously been used successfully in the design of video quality metrics [224, 227, 228], but the employed methods required a matrix or two-way array presentation of the information gained from the video sequences and thus temporal pooling was necessary.

In this thesis, I extend this approach by utilising multi-way data analysis methods that are able to handle the three-way nature of video without preceding pooling and thus consider the complete variation within a video sequence in the model building. In particular, I focus on data analysis methods that do not only consider the variation within the features but

also the variation within the visual quality during the model building process by introducing the multi-way partial least squares regression into the data driven design of video quality metrics. Furthermore, I also provide a comprehensive overview of two-way and multi-way data analysis methods, especially in the context of the design of video quality metrics.

High performance example metrics In order to evaluate if the data driven design with multi-way data analysis proposed in this thesis addresses the shortcomings of existing video quality metrics, I used the proposed approach to design two different example metrics, each representing one of the major categories encountered in state-of-the-art metrics. The first example metric represents universal or technology independent metrics and uses features that are extracted from the pixels contained within the video sequences. The second example metric represents technology specific metrics and in is based on features extracted from the bitstream of video sequences encoded according to the ubiquitous H.264/AVC standard. Both example metrics are no-reference metrics that do not require the undistorted video sequence for their quality prediction. As the focus in this thesis is on the data driven design approach, the two example metrics are not optimised manually towards specific test conditions, but are built using only multi-way data analysis without fine tuning.

The results in this thesis show that the proposed approach with multi-way data analysis outperforms the combination of two-way data analysis methods and temporal pooling. Moreover, even though purely data driven, the designed example metrics also outperform existing visual quality metrics in the state-of-the-art that are often highly optimised towards certain properties of the human visual system. In particular, the example metrics also fulfil all requirements discussed in the previous section unlike many metrics in the state-of-the-art. These results are based on a comprehensive validation using four different data sets spanning a large content, format and visual quality range, suggesting the validity of the results and trends in this thesis beyond the used video sequences.

1.3. Outline

This thesis consists of three distinct main parts as illustrated in Fig. 1.1 on the facing page: the first part provides an introduction into *video quality assessement*, the second part discusses *data analysis* methods, and the third part *design of video quality metrics* demonstrates how the data driven design approach with data analysis can be used in the design of video quality metrics. Following the three main parts, the appendix provides additional details and results, complementing the three main parts.

Part I – Video quality assessment In Part I, I focus on the question of what quality in general and video quality in particular is and how video quality can then be measured, both subjectively and objectively

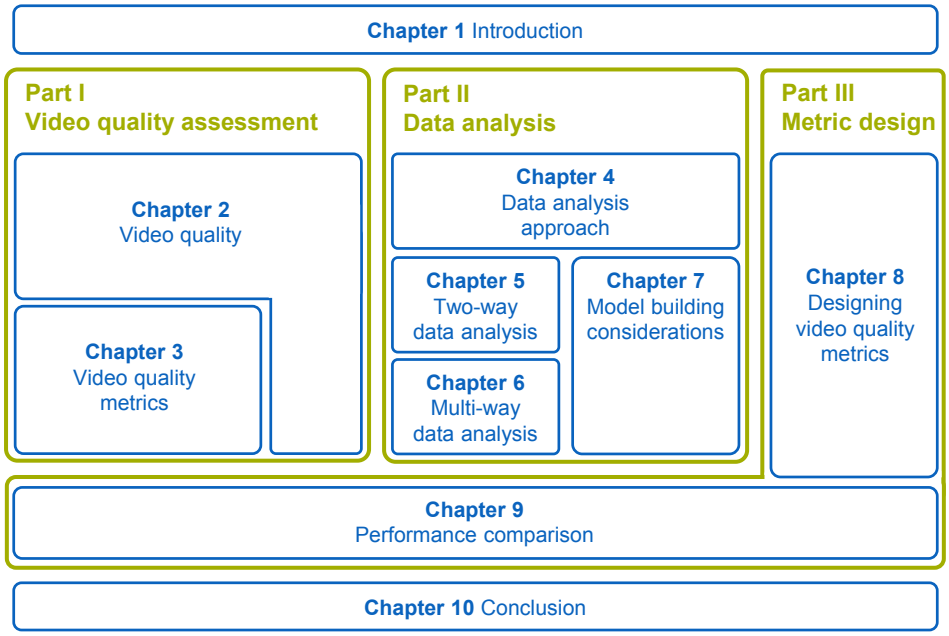

Figure 1.1.: Structure of this thesis

Following a review of existing quality concepts in the context of multimedia applications and services, I introduce in Chapter 2 the definition of video quality as used in this thesis that is based on the general concept of Quality of Experience (QoE). In light of this definition, methodologies and environments for subjective video quality assessment are discussed in detail that allow us to determine this video quality using human observers.

Based on this understanding of video quality and its assessment using human observers, I extend in Chapter 3 the scope of video quality assessment to video quality metrics, allowing us to evaluate video quality without human observers. Firstly, I define and justify the requirements for video quality metrics, and introduce a taxonomy of video quality metrics that allows us to categorise the different approaches to the design of video quality metrics. Using both requirements and the taxonomy, I then evaluate the state-of-the-art in video quality metrics and highlight the shortcomings of existing concepts in the design of video quality metrics.

Part II – Data analysis Part II is focused on the data analysis methods employed in the proposed data driven design approach to video quality metrics.

In Chapter 4, I provide an introduction to the general concept of model building with data analysis, followed by a discussion of the notation necessary to describe video sequences in an adequate form for the application of the data analysis methods, in particular with

respect to the three-way nature of video. The chapter concludes with the description of the preprocessing that should be applied before the data analysis is performed.

These preliminaries are followed by an in-depth discussion of traditional two-way data analysis methods in Chapter 5. After briefly reviewing temporal pooling methods that map three-way data to two-way data and that are necessary before two-way data analysis methods can be applied, I start the discussion on data analysis methods with multiple linear regression (MLR), before reviewing the concept of component models and latent variables. I then discuss two two-way component-based data analysis methods that can be used to extract latent variables from the data, the well-known principal component regression (PCR) and the lesser-known partial least squares regression (PLSR).

Using two-way data analysis methods as a foundation, I discuss in Chapter 6 first how these two-way methods can be applied to three-way data without temporal pooling, before introducing the concept of multi-way component models. Based on the well-established concepts of the Tucker3 and PARAFAC decompositions for multi-way component models, I provide a detailed discussion of the three-way extension of the PLSR, the trilinear PLSR, that is used in this thesis for building video quality prediction models. The discussion of the data analysis methods is concluded in Chapter 7 by addressing issues with respect to validation, model selection, component selection and feature selection that need to be taken into account in the data driven design approach with data analysis.

In the context of this thesis, the focus is on prediction *performance, not complexity*, and the algorithms for the different data analysis methods discussed in Part II may not necessarily be the computational most efficient ones. Similarly, the aim in this thesis is *prediction, not explanation*, and thus although all data analysis methods per definition allow for an interpretation of the features with respect to their influence on the predicted video quality, this option is not pursuit further in this thesis. Lastly, even tough the data driven approach can be implemented with many different methods, the focus in this thesis is on *analysis, not learning* and therefore alternative concepts to the data analysis approach e.g. machine learning or neural networks are not discussed.

Part III – Design of video quality metrics In Part III, I demonstrate how the data driven design approach with multi-way data analysis discussed in Part II can be used to design video quality metrics with a high prediction performance, providing results similar to the results gained in subjective quality assessment as described in Part I.

Chapter 8 introduces the two example metrics that are used in this thesis to evaluate if the proposed data driven design approach with multi-way data analysis is suitable for the design of video quality metrics. Firstly, I describe the technology-specific example metric based on features extracted from the bitstream of video sequences encoded according to the H.264/AVC standard, before describing the second example metric that represents universal or technology independent metrics, using features that are extracted from the pixels contained within the video sequences. Both example metrics are no-reference metrics that do not require the undistorted video sequence for their quality prediction. As the focus

of this thesis is on *design, not development*, the two example metrics are not optimised manually towards specific test conditions, but are built using only multi-way data analysis without fine tuning. Also I only focus on *features, not vision*, using only features extracted from the video sequences and do not rely on modelling the human visual system, following Winkler's remark in [354] that *the human visual system is extremely complex* and that *our current knowledge is limited*. Due to technical limitations of the bitstream-based example metric, only distortions caused by *coding, not transmission* are considered in this thesis.

The example metrics are then used in Chapter 9 for a performance comparison between the different data analysis methods, but also with visual quality metrics in the state-of-the-art. Before the results of the performance comparison itself are presented, the properties of the used performance metrics are discussed and the used data sets are introduced. The selection of the data sets was based on an evaluation of the publicly available data sets with respect to criteria defined in the context of this thesis. Following these preliminaries, I present the results of the performance comparison. Firstly, the different two-way and multi-way data analysis methods are compared for both example metrics with respect to the performance metrics by building a prediction model with each data analysis method for each example metric and data set. Based on this comparison, the best performing combination of data analysis method and example metric is selected and subsequently compared to various image and video quality metrics, representative of the state-of-the-art. The results of both comparisons are then briefly discussed in a short summary at the end of the chapter.

Finally, this thesis concludes in Chapter 10 with a brief summary. The comprehensive Appendix provides additional information to the data analysis methods, in particular the used algebra, but also additional results of the performance comparison and further information about some issues of video quality assessment and estimation.

Part I.

Video Quality Assessment

2. Video Quality

> Quality – you know what it is, yet you don't know what it is.
>
> ZEN AND THE ART OF MOTORCYCLE MAINTENANCE [252]
> *Robert M. Pirsig*

In video processing, often the subsequent question arises about the processed videos' quality and we face a dilemma similar to the protagonist in Pirsig's novel: we have a intuitive notion of the perceived videos' quality, yet find it difficult to provide an adequate description of its different aspects. Moreover, depending on the context and individual experience, different observers may arrive at different definitions of quality. Hence we need to agree on a generally accepted definition of video quality and corresponding methodologies that allow us not only to describe, but also to measure video quality. In this chapter, I therefore discuss the different concepts of quality. Starting with a general discussion on the possible definitions of quality, I review three quality concepts used in video processing: *Quality of Service (QoS)*, *Quality of Perception (QoP)* and *Quality of Experience (QoE)*. Based on these concepts, the definition of video quality used in this thesis is introduced, followed by an overview of methodologies and requirements to assess video quality subjectively.

2.1. What is Quality?

Quality. A word intuitively used every day, yet elusive in its meaning. Before defining video quality, it is therefore useful to first review the possible meaning of the word *quality* itself. As we are aiming to describe the quality of video, we are interested in the interpretation and definition of quality with respect to a specific object or entity, in our case represented by the individual video sequences. Martens and Martens [195] suggest that there are at least four different common definitions of the quality of an object:

Definition 2.1 – *Quality as Qualitas*
Quality is the essential nature, inherent characteristic or property of an object.

The first definition considers quality as an intrinsic property of an object, providing us with an objective meaning of quality. Relating this definition to video, quality is represented by the objectively measurable features of a video. But this understanding of quality does not decide if all the measurable properties are relevant or not.

Definition 2.2 – *Quality as Excellence*
Quality is any character or characteristic which may make an object good, bad, commendable or reprehensible.

The second definition addresses the implicit relevance of the objects' properties by taking into account that human observers are evaluating these properties. It assumes a common understanding of the *excellence* or *goodness* of an object, of what is good and bad about an object. Again considered from the video perspective, this definition corresponds to the subjective assessment of a video's excellence or goodness by a human observer.

Definition 2.3 – *Quality as Standard*
Quality is the ability of a set of inherent characteristics of a object to fulfil requirements of interested parties.

The third definition considers quality as the degree to which an object's inherent characteristics fulfil given requirements. It combines aspects of the first definition with the second definition: inherent characteristics of an object are defined as descriptive quality criteria that are then used as specifications to assess the object's excellence, where the assessment need not be done necessarily by human observers. Translating this definition to video quality, it corresponds to the assessment of video with respect to the degree that the excellence criteria as defined by certain values of the features are satisfied.

Definition 2.4 – *Quality as Event*
Quality is not a thing, it is the event at which awareness of subject and object is made possible.

The fourth definition considers quality as something that is not only dependent on the object itself, but also on the event or occasion in which the object occurs and is perceived. Hence this definition takes into account the context in which an object is observed. This definition of quality therefore considers the interpretation of an object's intrinsic properties as dependent on the context, resulting in a differently subjectively perceived excellence of the object in different contexts. Martens and Martens call it therefore also the *lived quality* in [195]. Expressed in terms of the above quality definitions, the *Quality as Qualitas* of the object leads to a different *Quality as Excellence* of the object depending on the context. For video quality, this definition can be understood to correspond to the subjective assessment of a video's excellence not not only depending on the video itself, but also depending on the viewing conditions and the human observers' constitution, representing the context in which the video is viewed.

Each of these four definitions of quality can in principle be used as a foundation of a possible definition of video quality, depending on which aspect of quality we want to focus on. But as our aim is to assess how differently processed or distorted videos are perceived and experienced differently by human observers, the first definition of *Quality as Qualitas* is not suitable, as it only describes the intrinsic, objectively measurable properties of a video, represented by the features. Similarly, the third definition of *Quality as Standard* is also not

suitable, as it uses only certain levels of the intrinsic, objectively measurable properties of a video to determine its goodness. Thus only the definitions of *Quality as Excellence* and *Quality of Event* are sensible foundations for a definition of video quality.

2.2. Quality of Service

Moving from the general consideration of the meaning of quality back to video processing, a commonly used quality concept associated with video quality is the *Quality of Service (QoS)* and one often used definition of QoS is given in the ITU recommendation ITU-T E.800 [118]:

Definition 2.5 – *Quality of Service (QoS*)
Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service.

Although aimed at telecommunications services, this definition can easily be adapted to our application area of video by substituting *telecommunication service* and *service* with *video*. Comparing this definition then to the general quality concepts in the previous section, QoS can be considered equivalent to *Quality as Excellence* in Definition 2.2, as it assess the excellence of the object's characteristics with respect to a certain need of a user i.e. a human observer. This definition suggests that QoS could therefore be an adequate concept to describe video quality, as it implies an evaluation of the video's excellence by human observers.

Unfortunately, QoS in practical use is quite different from the above definition and is focused exclusively on the evaluation of objectively measurable signals degraded by processing or distortions [173]. It can therefore be considered as a pure fidelity measure. Hence QoS in reality is equivalent to *Quality as Qualitas* in Definition 2.2 and thus not suitable for a definition of video quality. Depending on the application area and network type, different specifications for this interpretation of QoS exist as discussed in detail by Stankiewicz et al. [308].

MSE and PSNR - An example why QoS is not sufficient The inadequacy of QoS due its common interpretation as *Quality as Qualitas* for the description of *Quality as Excellence* can be demonstrated impressively on the example of the most popular QoS metric in image and video processing, the ubiquitous *mean squared error (MSE)* and the closely related *peak-signal-to-noise ratio (PSNR)*. Both provide a signal fidelity measurement with the average squared error between the original or unprocessed signal and the distorted or processed signal, where in our case the signal is represented by the pixels of an image or video frame.

Due to its simple definition and straightforward calculation, the MSE and PSNR are often used to describe video quality, thus equating QoS to video quality. The difference between distorted and undistorted image as represented by the MSE and PSNR, however, does

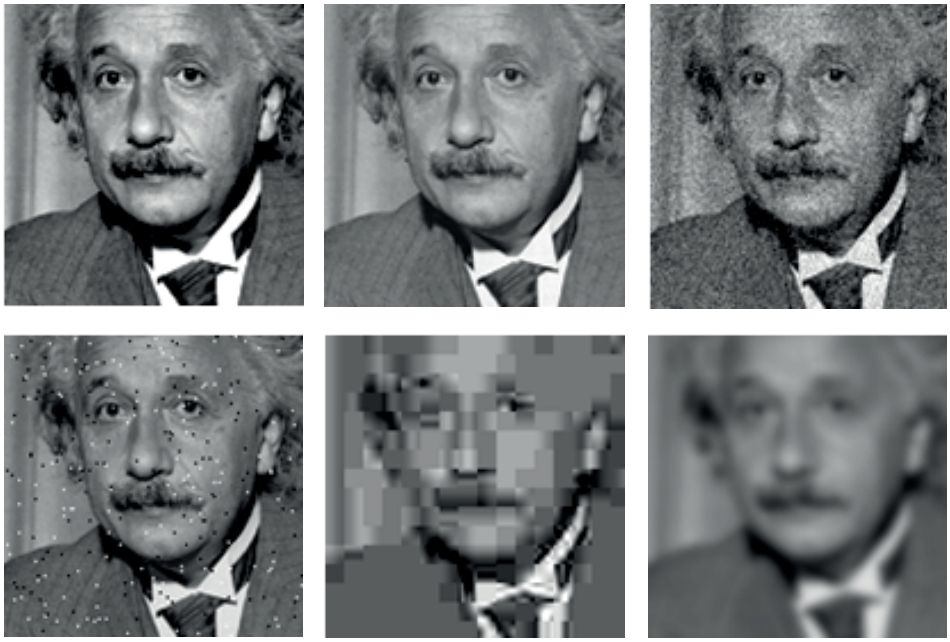

Figure 2.1.: QoS is insufficient: all images have the same MSE, yet clearly different visual quality depending on the influence of the distortion type on human perception (from [336])

not reflect the human quality perception adequately, as is illustrated in Fig. 2.1, where all distorted images have the same MSE, but clearly the perceived visual quality varies widely between the images. These shortcomings of the MSE and consequently the PNSR are well-known facts [73, 339, 368] and are discussed in detail by Wang and Bovik [336].

2.3. Quality of Perception

One concept less frequently used at least explicitly is the *Quality of Perception (QoP)*, representing the *Quality as Excellence* in Defintion 2.2. It addresses the shortcomings of QoS by using the subjectively perceived goodness of a distorted or processed video sequence as a description of its quality [55]. One possible definition of QoP is described in the ITU recommendation ITU-T E.800 by the *Quality of Service Experience (QoSE)* [118]:

Definition 2.6 – *Quality of Service Experienced (QoSE*)
A statement expressing the level of quality that customers/users believe they have experienced.

QoSE focuses on a service's *level of quality* as perceived by the human observer and is thus describing the subjective detectability of quality changes caused by processing or distortions [55]. Note, that even tough the above definition mentions *experience*, it focuses not on the overall experience itself but rather on a service's *level of quality* as the

experience. Using QoP as a definition of video quality, we can then express video quality as the *Quality of Excellence*.

Perception and consequently QoP, however, can not be separated completely from the context or event. Or expressed differently, can perceiving be separated from experiencing? Although the desired separation may be possible to a certain degree in the design of psycho-visual experiments by using randomised patterns to examine perceptual properties of the human visual system, the laboratory environment still provides a specific context. Considering the determination of the perceptual quality of video in general, each subjective evaluation is performed in a specific environment and using a specific methodology. QoP is therefore always including the context of the evaluation. Thus in practical use QoP is implicitly equivalent to an interpretation of video quality as *Quality as Event*, but does not explicitly consider the context in its definition as *Quality of Excellence*. Due to this ambivalence QoP is therefore also not a suitable definition of video quality.

2.4. Quality of Experience

Quality of Experience (QoE) complements the signal fidelity focused QoS and purely perceptual QoP by aiming to capture the *quality* as truly subjectively experienced by considering video quality as *Quality as Event* according to Definition 2.4. One popular definition of QoE is provided in the ITU recommendation ITU-T P.10/G.100 [122]:

Definition 2.7 – *Quality of Experience (QoE) – ITU-T P.10/G.100*
The overall acceptability of an application or service, as perceived subjectively by the end-user.
NOTE 1 - Quality of experience includes the complete end-to-end system effects.
NOTE 2 - Overall acceptability may be influenced by user expectations and context.

This definition considers the subjectively perceived quality with respect to the the human observer's expectations and context, thus it seems to provide a reasonable interpretation of *Quality as Event*. It does, however, define quality purely in the terms of acceptability, but following the argument by Möller [211], acceptability itself is based at least partly on the QoE. Therefore, this can not be a suitable definition of QoE.

In its Whitepaper on the Definitions of QoE [173], *Qualinet*, the *European Network on Quality of Experience in Multimedia Systems and Services*, provides a more holistic definition of QoE in the sense of *Quality as Event* by extending the definition of QoE beyond the pure acceptability as in the ITU-T definition [173]:

Definition 2.8 – *Quality of Experience (QoE) – Qualinet*
Is the degree of delight or annoyance of the user of an application or service. It results from the fulfilment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user's personality and current state.

The quality formation process resulting in a QoE rating according to Definition 2.8 is illustrated in Fig. 2.2 on the following page. Input parameters of this process are the context,

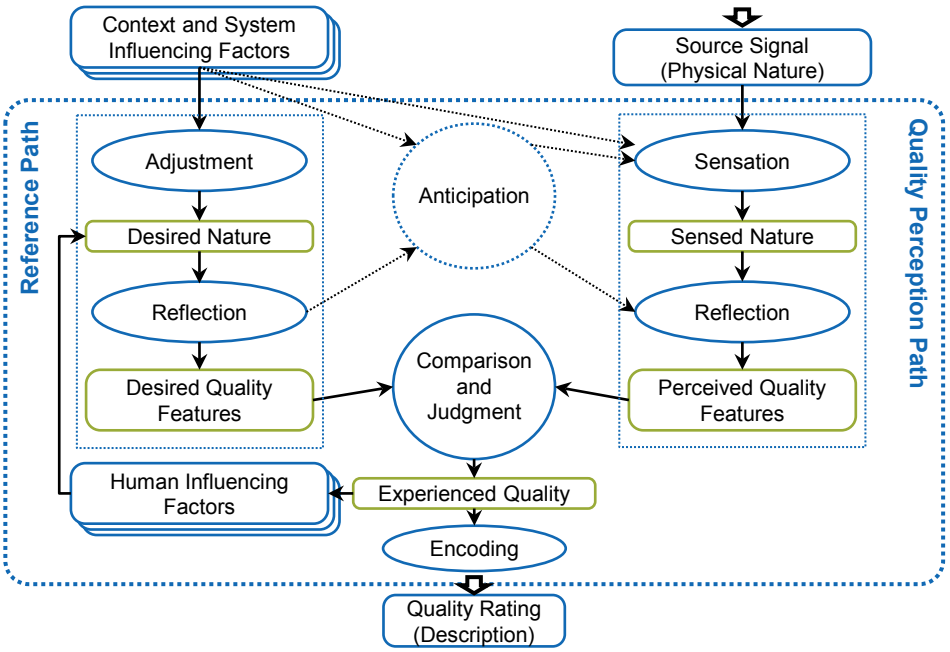

Figure 2.2.: Quality formation process resulting in a QoE rating: reference path on the left and quality perception path on the right (from [173])

representing the nature of the event both externally and internally, in our case with the viewing environment and the human observers' constitution, and the physical signal itself, represented in our case by the video. The quality formation part consists of two overall parts, a quality perception path, describing the sensation and perception of the visual signal, and the reference path that translates the context of the quality event into the expectation of a certain desired quality. The experienced quality is then gained by comparing the desired quality with the perceived quality, finally describing the quality of the complete event as QoE. Thus Definition 2.8 of the QoE provides a suitable interpretation of *Quality as Event*.

Moreover, we can observe that QoS and QoP can be considered as contributing aspects to the overall QoE in this definition: firstly, the objective properties of the input signal described by the *Quality as Qualitas* with QoS influence the sensing of the stimuli at the beginning of the perception path. Secondly, the perception path resulting in the perceived quality represents the *Quality as Excellence* as described by the QoP. The definition triple of QoS, QoP and QoE resulting from the quality formation process is also similar to the triple model for QoE with a sensorial, perceptual and emotional step as suggested by Pereira [245, 246]: the sensorial part is represented roughly by the QoS, the perceptual part by the QoP and the emotional part by the reference path, resulting in the overall QoE. Additionally, this process supports the argument in Section 2.3 that QoP is not a suitable definition of video quality, as we have no access to the result of the perception path itself

and thus the QoP: we can only observe the result of the quality formation process in the form of the QoE.

In practical use, however, the context is often not explicitly considered in the assessment of QoE, as QoE is usually assessed in formal subjective testing according to standardised methodologies and in standardised test environments. But it can be argued that this provides at least a standardised external context, even if the internal context of the subjects participating in a test may vary from subject to subject. Hence, if we consider the formal subjective testing as a sufficiently well-defined context and thus *event*, we can consider the results of these tests as a representation of the QoE for this specific context. Besides the definitions of QoE discussed above, alternative definitions have been proposed that explicitly include the business aspect e.g. Geerts et al. [71], Laghari and Connelly [166], Perkis et al. [247] and/or the influence of different demographics e.g. Geerts et al. [71], Laghari and Connelly [166] into the definition of QoE.

Based on these considerations and the *Qualinet* QoE definition in Definition 2.8, *video quality* in this thesis is therefore defined as a derivation of QoE:

Definition 2.9 – *Video Quality*
Is the degree of delight or annoyance of a subject in formal subjective testing, expressed by the resulting ratings on a specific scale provided to the subject. It results from the fulfilment of the subject's expectations with respect to the utility of the presented video sequences in the light of the subject's personality and current state.

On the one hand, this definition is less general than Definition 2.8, reflecting the focus on video and subjective testing, on the other hand, the enjoyment as one of the criteria for the subject's expectations is omitted, as in subjective testing the utility of the processed or distorted video is in the focus. Similarly, it is assumed that the subject's delight or annoyance is recorded on a specific scale provided to the subject and *formal subjective testing* in this context refers to the use of standardised testing methodologies and environments in the subjective testing.

Due to the fact that video in the context of video quality metrics is considered to only contain visual stimuli, *visual quality* is often used synonymously with *video quality* in order emphasise that the quality evaluation is limited to visual stimuli. In this thesis, I therefore use *visual quality* when referring to the subjective ratings and/or prediction results of the metrics in order to highlight the visual nature of the assessment and prediction task, and *video quality* when referring to the overall quality evaluation or metrics.

2.5. Video quality assessment

Based on my definition of video quality, this section describes the two aspects that define *formal subjective testing*, the viewing environment and the used subjective testing methodologies. *Formal* in this context means that both the used methodologies and the other test parameters are not designed especially for each test, but are rather based on standards or well-established best practices. The overall goal behind using a formalised testing setup is the elimination or at least significant reduction of possible biases introduced due to the test environment or methodology. A comprehensive list of possible biases in subjective quality assessment is given in the review by Zieliński et al. [376]. Although only focusing on audio, the assumptions behind many methods are similar to video and therefore the review also provides an insight into the biases that are likely to be encountered in video quality assessment. The secondary, but often equally important goal of a formalised testing setup is to ensure the reproducibility of the results across different testing sites.

In the planing of a subjective test, the processing or distortions to be studied and the media or applications to be targeted must be defined e.g. the evaluation of new coding technologies for certain resolutions or the assessment of the influence of wireless channel errors on video communication. Depending on the overall goal of a test, different testing methods are chosen, but also the testing environment may be adapted if needed.

2.5.1. Environment

The testing environment consists of the evaluation room and displays used to present the videos in the subjective testing. For both room and display properties, usually ITU-R recommendation BT.500 [109] is used as a guideline and depending on the overall test goal, modifications can be made on basis of this recommendation.

Room In order to avoid any unnecessary distractions from the visual assessment task for the subjects, the walls of the room are mid-grey as a colour-neutral compromise between a too dark or too bright background that could possibly conflict visually with the video shown on the display. Similarly, the test room should be sufficiently sound-proof to minimise distractions to the test subjects. Flicker-free, uniform lighting with a colour temperature of 6500 K equivalent to daylight provides the room's illumination and the lighting should be adjustable, so that a ratio of 0.15 between the peak illumination of the used display and the background illumination behind the display can be achieved. The seating of the test subject should allow for a variable distance between display and test subjects, depending on the used display's screen size. A typical example for a ITU-R BT.500 compliant video quality evaluation laboratory is shown in Fig. 2.3 on the next page. Pinson et al. [248], however, recently suggested in their comparative international study that uncontrolled environments may also be suitable for video quality assessment, as many factors kept constant in the controlled environment, e.g. lighting or wall colour, do not seem to influence the results significantly.

Figure 2.3.: ITU-R BT.500 compliant subjective testing laboratory at TUM: controlled lighting, colour-neutral mid-grey background, adaptability to different displays and test setups

Displays According to ITU-R BT.500 a reference monitor should be used for presenting the videos. Depending on the media or application under test, the display is calibrated to a specific colour gamut, gamma, white point and luminance. For the colour gamut, gamma and white point, ITU-R BT.601 [107] is used for standard definition television (SDTV) and ITU-R BT.709 [102] is used for high definition television (HDTV). The luminance should be 100 cd/m^2 to 200 cd/m^2 in compliance with ITU-R BT.500 and a reasonable choice is 120 cd/m^2 as recommended in SMPTE RP166 [301]. The viewing distance is depending on the display's screen size and expressed in terms of the screen height H. Common viewing distances are 3 H for HDTV [100] and 6 H for SDTV [101]. Instead of using an expensive reference monitor, a calibrated high-quality display can be a valid alternative as indicated by Keimel and Diepold [137].

Although ITU-R BT.500 still requires a CRT display for the video quality evaluation, both a study by VQEG [330, Appendix VII] and Pinson and Wolf [250] have shown that the results gained with LCD displays are statistically equivalent to the results gained with CRTs. Therefore LCDs are a valid contemporary choice to replace the increasingly rare and outdated CRTs. ITU-R BT.2022 [108] describes some additional considerations that should be taken when using non-CRT displays for subjective testing and the ITU-R recently published a draft recommendation for the viewing environment when using LCDs [114]. Pinson et al. [248] suggest additionally that unless the aim of test is in the assessment of different equipment, display calibration, display type and viewing distance may not have a significant influence on the results. Similar results were achieved in a smaller comparison of the results gained by performing a test with a reference display, a consumer display and a home cinema projector by Redl et al. [266].

Crowdtesting Crowdtesting describes the use of crowdsourcing to perform subjective testing. Instead of a dedicated video quality assessment laboratory, the subjective test-

ing is performed distributed in the Internet using web-based applications. One obvious advantage is a more demographically and geographically diverse group of subjects, more representative of the general population. From a more economically point of view, the costs associated with subjective testing can be lowered significantly using crowdtesting, as on the one hand the reimbursement of the test subjects can be lower and on the other hand less investment in equipment has to be made.

In moving the subjective assessment into the Internet, however, many parameters in the subjective testing are no longer fixed and may not even be controllable at all. First studies by Keimel et al. [140, 141] have shown that crowdtesting can indeed provide results similar to the results from formal subjective testing in a laboratory environment. But as also discussed by Keimel et al. [138], many challenges still remain before crowdtesting is a universally acceptable replacement for formal subjective testing.

2.5.2. Testing methodologies

Testing methods describe a set of certain aspects that define the test setup and process in detail. Often methods from the recommendations ITU-R BT.500 [109] and ITU-T P.910 [121] are chosen. Both suggest similar methods, but the former is focused on broadcasting, whereas the latter is focused on telecommunications applications. Instead of limiting this section only on the discussion of standardised methods, the aim of this section is rather to provide an understanding of the assumptions on which these standardised testing methods rely.

One important aim of these methods is to avoid the introduction of additional biases into the subjects' rating due to the assessment task. Considering the quality formation process in Section 2.4, one source for introducing biases is the encoding or mapping of the subjects' quality rating to the quality scale provided by the used methodology. This is not a problem unique to video quality assessment and can be observed whenever judgements need to be quantified. Poulton [260] provides a comprehensive general overview and discussion about the different types of bias that can be encountered in quantifying judgements, and Zieliński et al. [376] elaborates the biases discussed in [260] in the context of subjective assessment for auditory stimuli.

Content selection The used video sequences should cover a sufficiently large variety of content for the targeted media or applications and the processing or distortion under assessment should be able to produce noticeably degradation in the selected content. Even tough the content should be sensitive to degradations, it should still be realistic content, conceivably be part of real-life media and applications. This requirement is often expressed as *critical, but not unduly so* [109]. Additionally, the media and applications targeted in the test are usually also influencing the selection of the content. In the context of video quality assessment, an unprocessed video sequence containing a specific content is also often called *source*. For the majority of testing conditions, the video sequences are 10 s long

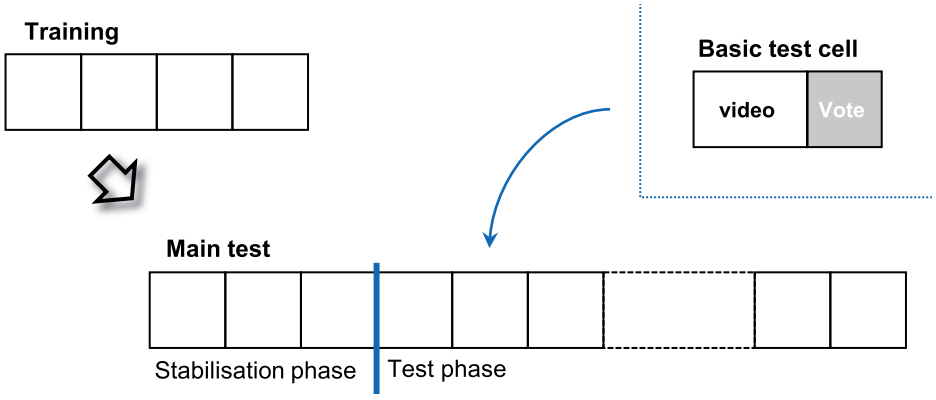

Figure 2.4.: General structure of a subjective test: training phase and main part, consisting of a stabilisation phase and the test phase itself.

to allow for a sufficiently large number of different test conditions within the test. Besides these economical considerations, the length of 10 s can also be justified by the fact that even if the video quality is assessed continuously, at most the last 9–15 s seconds of the video are considered by the subjects in the quality assessment as demonstrated by Pinson and Wolf [249] and also indicated by Aldridge et al. [1]. Some general criteria for the selection of adequate content are suggested in ITU-R BT.1210 [113] and by Pinson et al. [251].

General structure Each test consists of two major parts: a training and the test itself as illustrated in Fig. 2.4. Both consist of multiple basic test cells (BTC), each representing one specific test condition and including a separate block for recording the subjects' quality rating. The separation of assessment and voting allows the subjects to exclusively focus on the assessment task while the video is shown. The content and test condition of two successive BTCs should be different in order to avoid the transfer of the bias from the preceding to the current BTC caused by the similarity in content and test condition [260]. In order to avoid fatigue in the test subjects, the test itself should last no longer than 30 minutes and it may therefore be necessary to split larger test into different test sessions, where each session has the same overall structure as outlined above. Regarding the influence of the assessment task on the viewing behaviour of the subjects, a study by Le Meur et al. [174] indicates that the subjects' eye movements do not change significantly between the free viewing of the video without assigned task and the viewing of the video with the task to assess its quality.

Before the test starts, the *training* provides the test subjects with an introduction to the test setup and methodology. Also it offers the subjects an opportunity to practice the assessment task and, if necessary, ask for assistance from the test supervisor. The instructions are usually given by the test supervisor and it is sensible to use a prepared script

or presentation for these instructions, especially if the supervisor changes from subject to subject or the test is part of a joint campaign with other laboratories. In order to avoid influencing the subjects, the content in this training should be different from the test itself, but should still exhibit similar distortions as in the subsequent test. Should subjects be unable to perform the required assessment task they are consequently excluded.

After a successful training of the subjects, the test itself commences in two phases: first a stabilisation phase, followed by the test phase.

The *stabilisation phase* provides the subjects with an indication of the range of visual quality they will encounter during the test. The aim of providing quality anchors is to reduce or at least control the absolute contraction, centring and range equalising bias. Contraction bias describes the avoidance of the test subjects to use the extremes of the scales and thus a contraction of the subjects' ratings in the direction of the scale centre. The centring bias describes the tendency of subjects to shift their quality range towards the centre of the assessment scale, so that their ratings a symmetric towards the scale's centre. Lastly, the range equalising bias describes the effect that even if the subjects only use a small range of their inner quality scale, they map it to the complete range of the assessment scale [260].

By providing anchors for the complete quality range in the test including best and worst quality, the subjects are able to adjust their inner, intrinsic quality scale to the provided quality scale [260]. This reduces the contraction bias, as the subjects are now familiarised with the quality range they will encounter in the test, and controls the centring and range equalising bias as all subjects will now share the same intrinsic quality scale. Usually the stabilisation phase consists of three to five video sequences [109]. Because the subjects are not aware of this implicit stabilisation phase, this provides an *indirect anchoring* [376]. The ratings gained from the subjects in the stabilisation phase are discarded before processing. If the test consists of multiple sessions, the study by Keimel et al. [147] indicates that the test conditions in the stabilisation phase of each session should be representative of the quality range in the complete test, not only of the quality range in the current session.

Following the stabilisation phase, the *test phase* itself commences. It consists of the BTCs representing the test conditions that should be assessed with respect to their video quality. For each BTC, a corresponding rating is recorded that provides the subjects' ratings.

Subjects Subjects participating in a test should be screened for normal (corrected) visual acuity and colour vision using e.g. Snellen and Ishihara charts for vision acuity and colour vision, respectively [109]. Although it is recommended to reject subjects failing these vision tests, Pinson et al. [248] recently suggested that slightly less than perfect visual acuity and colour vision do not seem to influence the quality assessment significantly.

Usually non-expert or naïve viewers are preferred. Non-expert in this context means that the subjects were not involved in defining the processing or distortions introduced in the test conditions and therefore have no preconceptions about the degradations of the video

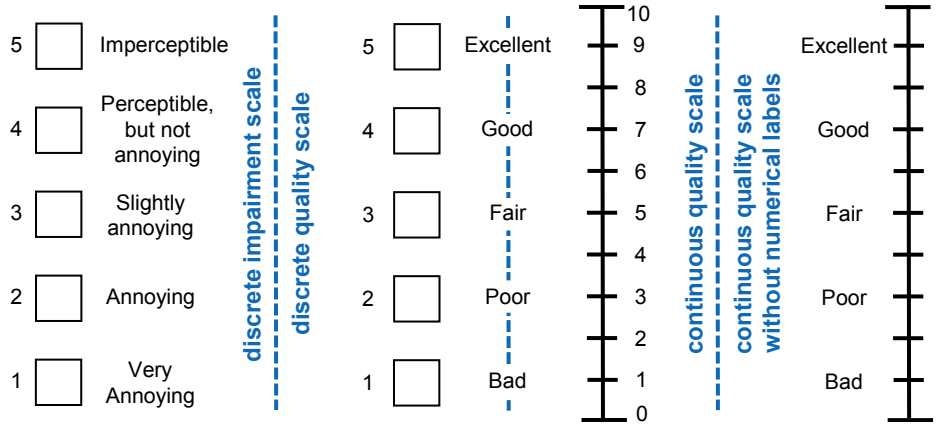

Figure 2.5.: Examples of rating scales used in subjective testing: discrete impairment scale, discrete and continuous quality scales with categorical and numerical labels as used in ACR [121], and continuous quality scale with only categorical labels as used in DSCQE [109]

quality caused by specific distortions [109]. The assumption is that non-expert viewers are therefore more representative of the general population than experts. Experts, however, may be used in the design of the test in order to choose appropriate test conditions in a pilot test [121].

ITU-R BT.500 and ITU-T P.910 recommend to use at least 15 subjects in the subjective testing. Winkler [351] confirmed this minimum number of 15 subjects in simulations and based on the data from five different experiments, but Pinson et al. [248] recommend based on the results of their comprehensive study that at least 24 subjects should be used in a controlled environment, and at least 35 subjects should be used in an uncontrolled environment. Regarding the use of expert or non-expert viewers, Nezveda et al. [219] have suggested that when using expert viewers, fewer subjects are needed compared to using non-expert viewers in order to provide similar results. They caution, however, that using only comparably fewer expert viewers may only be suitable to identify general trends.

Rating scales Rating scales allow the test subjects to record their quality assessment for each of the BTCs. Depending on the aim of the test, discrete or continuous scales may either record the impairment or the absolute quality of the video sequences under test. In addition to indicating the range of the scale, corresponding labels for certain impairment or absolute quality categories are provided on the scale and depending on the method the scale's range is often also indicated with numerical labels. Usually the scales are divided in five-, nine- or eleven-point intervals, but still usually only five categorical labels are used. Examples of discrete and continuous scale are shown in Fig. 2.5.

Although the required scale is often explicitly defined in the corresponding standard for the used testing method, the used scales are often adapted depending on the test setup.

Continuous scales consist of a line with ticks in equally sized intervals and categor-

ical labels indicating the position of certain impairment or absolute quality levels within the used scale. Often the ticks also provide additional numerical labels to indicate the available impairment or quality range. Even tough a continuous scale suggests that more granular ratings could be achieved, quantisation effects usually occur, as test subjects tend to align their ratings with the labels and ticks, resulting in a quasi-discrete distribution of the ratings [376]. In the context of video quality, a study by Huynh-Thu et al. [94] and an eye-tracking experiment by Schleicher et al. [280] confirmed this behaviour.

Discrete scales are similar, but unlike a continuous scale, only discrete choices are available to the subjects: each discrete option is represented by a box with a categorical and numerical label indicating the corresponding impairment or quality level. Considering the quantisation effect for continuous scales, the difference in the ratings resulting from using discrete or continuous scales are often negligible as indicated in the two following studies: Huynh-Thu et al. [94] presented results for the single stimulus ACR method, indicating that for both discrete and continuous scales with absolute quality categorical labels there are no statistically significant differences in the subjects' ratings. Additionally, no difference for scales with five, nine or eleven ticks was found. For the double stimulus DSCQS and DSIS II methods, Corriveau et al. [47] also presented results that indicate equivalence not only between continuous and discrete scales, but also between impairment and quality scales. In addition, Svensson [316] suggests that discrete scales provide better stability with respect to the intra-rater agreement in different tests.

Especially in a larger, international test, a potential issue for both types of scales is the position of the categorical labels with respect to the range represented by the corresponding scale, as depending on the impairment or quality, the labels' position may not be a suitable representation of the labels interpretation in specific languages [345, 376]. This was confirmed for impairment and quality labels commonly used in video quality assessment in a study by the ITU-R [99]. Despite this, the recent study by Pinson et al. [248] suggest that for English, French, German and Polish differences in the interpretation of the labels do not influence the overall results significantly.

Single stimulus methods Single stimulus methods consist of BTCs with a straightforward structure: each BTC consists of one test condition, followed by a voting block indicating the number of the current BTC and presented on a neutral mid-grey background as illustrated in Fig. 2.6 on the facing page. This simple structure allows for short BTCs and with a video sequence length of 10 s, followed by a voting block of 5 s, each single stimulus BTC lasts 15 s. Hence, we are able to achieve 4 BTC/min and with a maximum test duration of 30 minutes, we can therefore present 120 different test conditions per test.

One disadvantage of single stimulus methods, however, is their context dependency as they suffer from the sequential contraction bias: if the preceding test condition represented a high quality stimulus, the magnitude of the current stimulus' quality is overestimated, and if the preceding test condition represented a low quality stimulus, the magnitude of the current stimulus' quality is underestimated [260]. In order to avoid this, each test condition

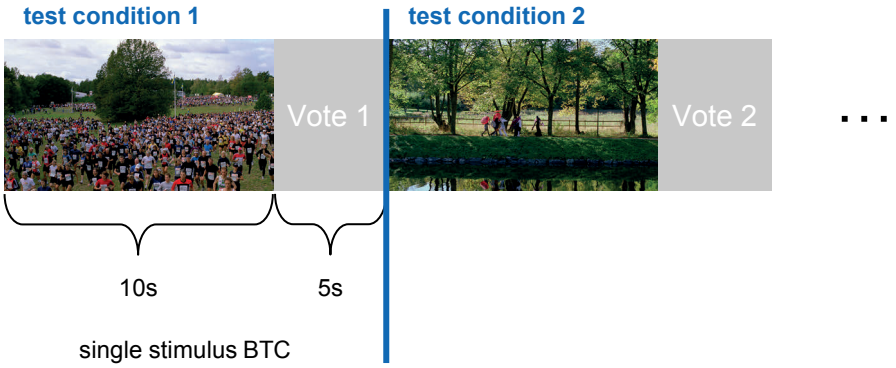

Figure 2.6.: Single stimulus BTC: presentation of the test condition, followed by a voting block

is shown multiple times during the test session in different context i.e. a different preceding test condition, and ITU-R BT.500 suggests three repetitions of each test condition in a test session. The rating for a test condition is then determined by averaging the results over all presentations, where ITU-R BT.500 also recommends to consider the first presentation as a stabilisation and therefore discard it in the averaging. In contrast, ITU-T P.910 suggest only two to four repetitions in total, but not for each test condition. Note, that with these repetitions the number of different test conditions per test is reduced accordingly. As a side effect, the multiple ratings of the same test condition can be used to assess the consistency of a subject's rating as an indicator of their reliability [121].

In ITU-R BT.500 [109], two basic variants are suggested: variant I, the *single stimulus (SS)* method, and variant II, the *single stimulus with multiple repetition (SSMR)* method, where the later includes multiple repetitions of each test condition to minimise the context effect. Both variants are frequently used with a discrete five-point impairment or quality scale with categorical and numerical labels, unofficially called *single stimulus impairment scale (SSIS)* e.g. in [205, 228] and *single stimulus multimedia (SSMM)* e.g. in [225, 346], respectively. Although not specified in ITU-R BT.500, often discrete nine- or eleven point quality scales are used e.g. a eleven-point scale for the SSMM by Oelbaum et al. [225]. Less frequently used extensions of these types are the *single stimulus numerical categorical scale (SSNCS)* that uses a discrete eleven-point scale with only numerical labels, and the *non-categorial scale* that uses a continuous scale without any labels.

In ITU-T P.910 [121], the *absolute category rating (ACR)* method is described. It uses a discrete absolute quality scale with categorical and numerical labels. Depending on the required discriminability, five-, nine- or eleven-point scales may be used, but also an option to use a continuous scale is provided in ITU-T P.910. Similar to the SSMR method, multiple presentations of each test condition are advised for the ACR method in ITU-T P.910. A derivation of the ACR method is the *absolute category rating with hidden reference (ACR-HR)*, where an undistorted reference version of each content is included in the

Figure 2.7.: Double stimulus BTC: presentation of reference *A* and test condition *B*, repeated once and then followed by a voting block

test, unknown to the subjects. For each subject, the rating of the hidden reference is then used to calculate a differential rating between each test condition and the corresponding undistorted reference with the same content. This can be considered as a rough correction of subjects' biases in the ratings at least with respect to the upper end of the provided quality scale.

Double stimulus methods Double stimulus methods extend the BTCs of the single stimulus methods with a *reference*, representing an undistorted version of the same content as in the test condition. Additionally, both the video representing the test condition and the reference video are usually repeated as illustrated in Fig. 2.7. This allows the subjects to gain an overall impression in the first presentation, followed by a detailed consideration of their rating in the second presentation. The video sequences representing the test condition and the reference are denoted with the letters *A* and *B*, respectively. Depending on the used method, the reference is either identified explicitly to the subjects or not. The order of the reference and the test condition is the same in each presentation, but each BTC may have a different order. Similar to the adaptation of the rating scales, both the order and the explicit identification of the reference are often adapted as needed.

One advantage of double stimulus methods is that the explicit reference in the BTC can be considered as *one-sided direct anchoring* [376]. Hence we can avoid on the one hand the sequential contraction bias that occurs in the single stimulus methods, on the other hand we provide a direct high quality anchor in each BTC comparable to the stabilisation phase, allowing the subjects to adjust their inner quality scale at least to the upper end of the provided quality scale in each BTC anew. Although multiple replications of each test condition are therefore not necessary, ITU-T P.910 suggests also for double stimulus methods two to four repetitions in total for reliability testing of the subjects [121]. Corriveau et al. [47] have provided experimental evidence that for double stimulus methods, in partic-

ular the DSCQS method, the context dependency is indeed significantly lower compared to single stimulus methods.

One practical disadvantage, however, are larger BTCs: assuming again a video sequence length of 10 s, a voting block length of 5 s and a *A*/*B* label block of 2 s, each double stimulus BTC lasts 53 s. Hence, we are able to achieve approximately 1 BTC/min and with a maximum test duration of 30 minutes, we can therefore present only 30 different test conditions per test, compared to up to 120 test conditions per test for the single stimulus methods.

In ITU-R BT.500 [109], two double stimulus methods are described: the *double stimulus impairment scale (DSIS)* method and the *double stimulus continuous quality scale (DSCQS)* method. The DSIS method uses a discrete five-point impairment scale in two variants: variant I that consist only of one presentation of test condition and reference, and variant II that consists of two presentations. In a BTC of the DSIS method, the reference is always the first video i.e. *A* and the test condition is the second video i.e. *B*. Moreover, this fixed order is announced to the subjects. Variant I of the DSIS method is also defined as the *degradation category rating (DCR)* method in ITU-T P.910 [121]. The DSCQS method is aimed at assessing the difference in quality between two videos. It uses two continuous scales with categorical labels and ticks in equal intervals for each BTC. Unlike the DSIS method, the subjects are not aware which video is the reference and they have therefore to provide separate ratings for video *A* and *B*, resulting in a differential rating $A - B$ for each BTC.

Baroncini [12] argues that this repetitive dual rating task in the DSCQS method leads to increased fatigue in the test subjects and therefore suggests an alternative to the DSCQS method based on a modification of the DSIS, variant II method, the *double stimulus unknown reference (DSUR)*. Unlike the DSIS method, the subjects are unaware if the reference is the first or second video. In the first presentation, the subjects are therefore asked to identify if *A* or *B* is the reference, and only in the second presentation they should then rate the non-reference video. Thus no repetitive task as in the DSCQS method is required of the subjects, but rather two different, though still related tasks must be performed by the subjects. In [12] a five-point discrete impairment scale with categorical and numerical labels is proposed, but it is often used with an eleven-point discrete quality scale e.g. in [225].

Other Methods *Continuous quality evaluation* methods aim to assess temporal quality changes or fluctuations by presenting the subjects longer video sequence of up to five minutes and allows them to assess the quality continuously using a slider. Both single stimulus and double stimulus methods are defined in ITU-R BT.500 [109]: the *single stimulus continuous quality evaluation (SSCQE)* method and the *simultaneous double stimulus for continuous quality evaluation (SDSCE)* method. For the SSCQE, only the distorted video sequence is presented, whereas for the SDSCE both the distorted and the reference video sequence are presented simultaneously on a shared display or two different displays.

The interactive *Subjective Assessment of Multimedia VIdeo Quality (SAMVIQ)* method defined in ITU-R BT.1788 [104] allows the test subjects to assess different test conditions of each content as often as needed in an interactive interface, including the undistorted reference for comparison. The rating itself is performed on a continuous quality scale with categorical and numerical labels from 0-100. Péchard et al. [239] have shown that SAMVIQ provides comparable results to ACR, but with fewer subjects. Due to its interactive approach, however, a SAMVIQ based test takes more time compared to an ACR based test.

Pair comparison methods are an alternative to the methodologies discussed so far. Instead of quantifying judgements, the subjects compare in each step two test conditions and decide which one is the preferred or better one. As the subjects do not need to quantify their judgements on a scale, we avoid the biases introduced by the mapping of the internal quality to the provided scale. By estimating the probability of choosing a test condition from the conditional probability that a certain test condition is preferred against another test condition with maximum-likelihood estimation, we can then use the Bradley-Terry-Luce model to obtain the rating of a test condition based on its probability [20]. One general disadvantage of the pair comparison, however, is that each test condition needs to be compared with every other test condition and assuming N different test conditions, this leads to $N(N-1)$ pair comparisons, which leads to significantly more BTCs than for the other methodologies. The Pair comparison variant as described above is recommended as *pair comparison (PC)* method in ITU-T P.910 [121]. ITU-R BT.500 [109] additionally suggests variants that also include a scale to quantify the magnitude of the difference, but this may reintroduce biases into the ratings.

Processing of the results After completing the test, the ratings of all subjects are averaged for each test condition, resulting in a test condition's *mean opinion score (MOS)*, representative of its video quality. If only difference ratings are available e.g. for the ACR-HR method, the average is called *differential mean opinion score (DMOS)*. As an indication of the MOS' uncertainty, usually the 95% confidence interval is additionally provided for each MOS.

In order to identify unreliable subjects, it is suggested in ITU-R BT.500 and ITU-T P.910 to perform a screening of the subjects according to a statistical criterion and reject those subjects' ratings that fail the screening. Pinson et al. [248], however, argue that only subjects that didn't understand the assessment task should be eliminated as it is often unclear why different subjects respond differently to the test conditions and therefore a strict statistical outlier removal may not be suitable. Still, outlier screening is usually done in subjective testing and two often used statistical criteria are discussed briefly in the following paragraphs.

ITU-R BT.500 [109] proposes to screen the subjects by comparing the rating of each subject for a test condition to the MOS of this test condition. If a subject's rating for a test condition deviates more than the sample standard deviation for this test condition multiplied

by a factor that is dependent on the ratings distribution, counters are incremented, one if the subject's rating is too low, another if a subject's rating is too high. These counters therefore provide not only the number of rejected ratings, but also indicate the reason for the rejection. Should both the number of rejected ratings for a subject be above a certain threshold and the ratings are not exhibiting any recognisable offset, the subject is completely rejected and all its ratings discarded.

VQEG [330, Appendix V] suggests a simpler method by calculating first the Pearson correlation coefficient r_p between a subject's ratings and the MOS for all test conditions, followed by a comparison with a threshold. If $r_p < 0.75$, the subject is rejected and all its ratings are discarded.

3. Video Quality Metrics

Video quality can easily be determined by formal subjective testing with one of the many methods discussed in the previous chapter. Subjective testing, however, not only needs to be organised by recruiting subjects and preparing the test setup, but also the execution of these test must be supervised. This is time consuming and often expensive. Moreover, it is clearly not possible to perform a real-time video quality evaluation of live content being broadcast or streamed.

Video quality metrics solve this problem by replacing the subjective testing with *models* that allow as us to predict the subjective visual quality from objectively measurable properties of the video sequences under test. These properties of the videos are called *features* and each feature represents a certain property. A video quality metric then utilises a set of different features in a prediction model to provide an estimation of the video quality. Typical features are *blur* or *blockiness*, describing how strongly details have been lost e.g. due to the loss of high frequency components in quantisation and how visible blocks are e.g. due to block-based transformations in video coding, respectively. One aim of video quality metrics is obviously to deliver quality predictions that are equivalent to the results form subjective testing, but additionally also other conditions may be imposed on the design of a video quality metric in order to ensure that this prediction performance can be achieved in all situations. In this chapter, I therefore introduce also the requirements that a video quality metric should fulfil and provide a classification scheme of video quality metrics. The requirements and classification are then used to review the state-of-the art in video quality metrics.

It should be noted, that Richter [268] suggests that video quality metrics should not be called metric at all, as they are not metrics in the strictest mathematical sense, but should be rather called *video quality indices*, as they are providing a relative ordering of video sequences with different quality for a given assessment algorithm. This distinction, however, has so far not caught on in the research community and therefore I use in this thesis the established term of video quality metrics.

3.1. Requirements on video quality metrics

Before we can design a video quality metric, we need to define the requirements a metric should fulfil in order to be considered adequate for replacing subjective testing. In this thesis, the requirements on video quality metrics are defined with respect to four categories: *temporal pooling*, *prediction performance*, *validation* and *reference availability*.

The requirements, however, are not only essential in the design process of our own metric, but also in the assessment of other metrics. In order to allow for a more gradual assessment, three levels of compliance are introduced for the requirements in each category: *full compliance*, fulfilling the requirements in the category fully, *acceptable compliance*, fulfilling the requirements in a category at least partly to an acceptable degree, and *insufficient compliance*, if the requirements are not met. An overview of the different compliance levels for the different categories is provided in Table 3.1 on the facing page. Clearly, an overall acceptable metric should fulfil the requirements in all categories fully.

Requirements on temporal pooling The temporal nature of video is not only an important issue in the design of video quality metrics because of the intrinsic temporal sensitivity of the human visual system as expressed by the spatio-temporal filtering of perceived stimuli, but also because the quality assessment task itself is influenced by the temporal nature of video. Thus not every frame of a video is considered separately, but rather the quality rating formation is an integral process occurring over a time window.

Psychological studies suggest that the global evaluation of an (affective) experience is mainly influenced by the stimuli's peak intensity and the intensity at the end of the experience [68], and that although the duration of the affective experience itself does not influence the overall experience [69], the overall trend of the intensity change towards the end has an influences on the overall experience [9]. This is supported by the results in [89] that suggest that the rating of the experience depends on the response mode i.e. if the evaluation is performed continuously or only globally at the end.

More specific to the context of video quality evaluation, Pinson and Wolf [249] suggest that for a quality rating at a given point in time, only up to 15 s preceding this moment are taken into account by the observers in the quality formation process. Moreover, studies by Moorthy et al. [214] and Seshadrinathan and Bovik [283] indicate that in subjective testing the simple averaging of multiple sub-sequence MOSs gained by continuous assessment does not represent the overall sequence MOS gained by the assessment of the complete video very well. Also Hands and Avons [80] suggest that the overall quality rating is strongly influenced by the observed peak impairment intensity during the assessment. Hence this suggests that the overall subjective quality of a video sequence can not be simply gained by pooling a set of multiple, more granular ratings.

Considering that for image quality metrics the quality prediction of a single frame is similar to a sub-sequence MOS gained in continuous assessment, it stands to reason that averaging over the quality predictions for each frame is as unrepresentative in video quality metrics as it is in subjective testing. This is assuming that the quality prediction is only based on the current frame itself and of course that the image quality metrics' quality predictions can are equivalent to the MOS from subjective testing. Furthermore, one can argue that for video quality metrics that consider the temporal nature of video only in a relatively small interval of frames around each frame and then gain an overall video quality prediction by a pooling with averaging over all frames, a similar effect might occur.

Table 3.1.: Compliance levels for the different requirements categories

Compliance level		Description
Temporal		
●●	full	no pooling
○●	acceptable	pooling other than averaging or at least consideration of preceeding frame
○○	insufficient	averaging
Reference		
●●	full	no-reference
○●	acceptable	reduced-reference
○○	insufficient	full-reference
Prediction		
●●	full	$r_p \geq 0.90$
○●	acceptable	$0.85 \leq r_p < 0.90$
○○	insufficient	$r_p < 0.85$
Validation		
●●	full	multiple data sets and cross validation
○●	acceptable	large data set and cross validation
○○	insufficient	small datset (less than 5 videos) or no cross calibration

In general, video quality metrics should therefore consider the temporal nature of video appropriately or if the quality predication is performed frame-by-frame at least the applied temporal pooling of the individual frames' quality prediction into one overall quality value describing the complete video should properly account for the temporal nature of video. That averaging is not the best pooling method for video quality metrics, has been demonstrated for different metrics in my previous contributions [143, 145], but also better pooling methods are inferior to the complete consideration of all frames in the quality prediction as demonstrated in [142].

For full compliance with this requirement, metrics should therefore consider all frames in the video or at least a sufficiently long scene of the video in their quality estimation process. For acceptable compliance, metrics that provide a frame-by-frame quality estimation and then pool over all frames should either at least consider the preceding frame in their quality prediction before averaging over all frames or use more sophisticated pooling schemes. If the quality prediction is performed frame-by-frame and then just averaged over all frames, the compliance with this requirement is deemed insufficient.

Requirements on reference availability Video sequences pass usually through a processing and transmission chain before they finally arrive at the viewers' display and both processing and transmission can introduce distortion in the videos. These distortions may

be as simple as artefacts due to the in-camera compression of the video or the inherent sensor noise of any camera, but can also include distortions introduced by transcoding in the transmission chain or transmission errors.

Video quality metrics should be able to provide a quality prediction at any point in the chain and should therefore not need additional information besides the video itself. In particular, they should not require the undistorted reference for their quality predictions as in real-life applications it is in the majority of cases not possible to provide the reference at the endpoint of the processing chain.

For full compliance with this requirement, metrics should therefore be *no-reference* metrics and not require the reference for their quality predication. For acceptable compliance, metrics should only require a description of the reference with meta-data and not the reference itself, and therefore be *reduced-reference* metrics. If a metric is a *full-reference* metric and requires the undistorted reference for the quality prediction, the compliance with this requirement is deemed insufficient.

Requirements on prediction performance Video quality metrics should be able to provide a quality prediction equivalent to the results from subjective testing. The prediction performance of video quality metrics is in most contributions expressed using the Pearson correlation coefficient r_P between the quality prediction of the metric and the MOS from a subjective test. The larger r_P, the better the prediction performance of the metric and a value of $r_P \geq 0.9$ usually indicates a very good prediction performance.

For full compliance with this requirement, metrics should therefore have a $r_P \geq 0.9$, for acceptable compliance $0.85 \leq r_p < 0.90$ and if $r_p < 0.85$, the compliance with this requirement is deemed insufficient.

Requirements on validation Directly related to the issue of prediction performance, is the issue of validation, as the assessment of the prediction performance is based on the validation of the metrics with a dataset of videos and corresponding MOSs. In order to provide a realistic assessment of the prediction performance, the used dataset should cover a sufficiently large number of different representative content and formats. Additionally, if the metric requires a training with video sequences and corresponding MOSs to generate a prediction model, a cross validation should be used.

For full compliance with this requirement, metrics should therefore use multiple data sets or at least a large data set with multiple formats in the validation and, if applicable, perform a cross validation. For acceptable compliance, metrics should use a large dataset with an adequate representation of different content and, if applicable, perform a cross validation. If a metric only uses a small data set with less then five videos with different content of and, if applicable, no a cross validation was performed, the compliance with this requirement is deemed insufficient.

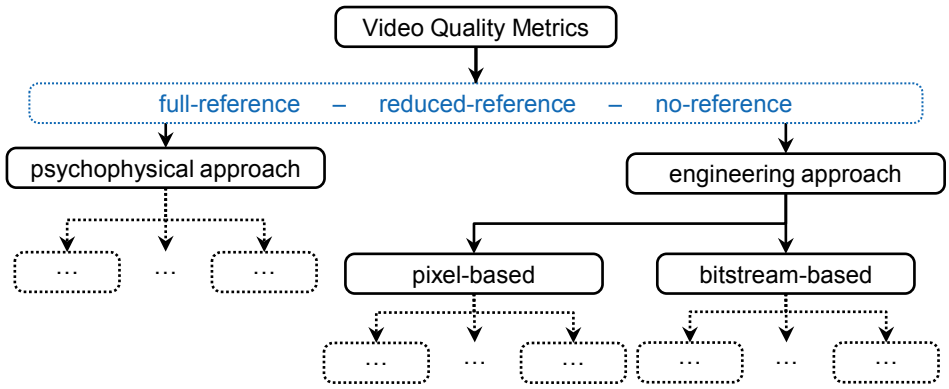

Figure 3.1.: Taxonomy of video quality metrics: vision versus features, universal versus specific and reference versus no-reference

3.2. Taxonomy of video quality metrics

Video quality metrics can be classified according to different criteria, regarding their general design approach, the availability of a reference and the scope of their applicability. Each of these criteria is discussed briefly in this section and an overview of the taxonomy is illustrated in Fig. 3.1.

Modelling vision or using features Winkler [350] differentiates between the *psychophysical approach* and the *engineering approach*, also called *model-based* and *signal-driven* by Lin and Kuo [181], respectively. The *psychophysical* approach describes metrics that a primarily based on modelling the human visual system (HVS). This can be considered as the implementation of an artificial observer with properties of the HVS. In contrast to this, the *engineering approach* describes metrics that are based on using a set of features extracted from the video. Each feature is assumed to describe a certain aspect of the video either with respect to its structure or the occurrence of certain artefacts and often they are based on specific psychophysical properties of the HVS.

But even though some features are based on the HVS, the main difference between these two approaches according to Winkler [350] is that the psychophysical approach aims at using a fundamental model of vision, whereas the engineering approach aims at analysing the the video. Although sometimes algorithms describing only the extraction process of a single feature are already considered as a video quality metric, this is not considered as a metric in this thesis.

Universal or technology-specific Metrics designed according to the engineering approach, can be distinguished into two major categories: *universal* or *pixel-based* and *technology-specific* or *bitstream-based*. The difference between these two categories is

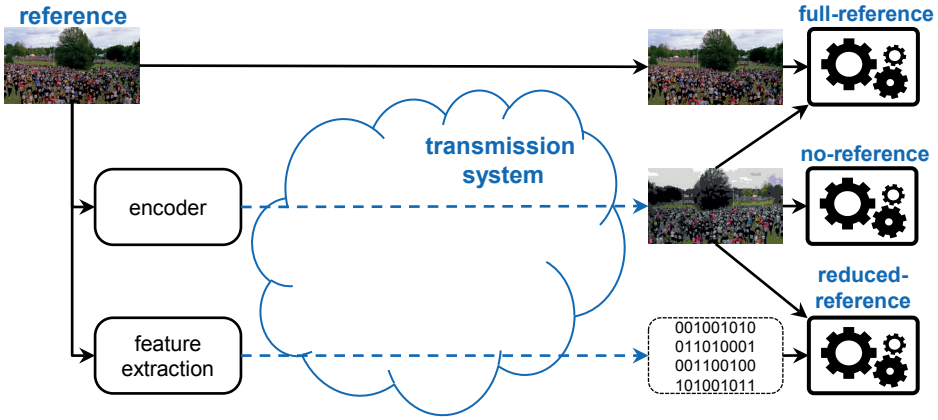

Figure 3.2.: Full-reference, reduced-reference and no-reference metrics in the context of a transmission system

illustrated on the case of video coding technology, but can also be extended to other technologies e.g. image sensor characteristics.

Pixel-based metrics only require the (decoded) video as input and are therefore independent from the used coding technology. Thus they are universally applicable and not limited to the quality assessment of encoded video, but also allow the assessment of processing algorithms in general.

In contrast, bitstream-based metrics require a certain technology dependent bitstream, representing the encoded video. Hence, they are custom-fit to distortions either introduced by the encoding itself or by the influence of transmission errors on the bitstream. On the one hand, this severely limits the universal applicability of bitstream-based metrics, but on the other hand this limited scope also allows for a much better adaptation to a specific range of distortions, resulting usually in higher prediction performance for bitstream-based metrics. Considering, however, that (standardised) coding technologies are usually widely adopted, the focus on a specific technology does not necessarily limit its applicability in practical applications. For both categories, further sub-categories can be defined describing the use of certain features or methodologies in the design of the metrics.

Reference or not Lastly, video quality metrics can be classified with respect to the information required about the un-distorted reference into either *full-reference*, *reduced-reference* or *no-reference* metrics as illustrated in Fig. 3.2.

Full-reference metrics require the un-distorted reference video in addition to the distorted video as an input for the quality prediction. Hence, they can compare the features in an un-distorted and distorted version of the video and use this information to provide a video quality estimation. Clearly, this limits the practical applicability, as often the undistorted reference is not available e.g. in transmission systems, where the videos are encoded exactly because it is often not feasible to transmit the not encoded reference.

Reduced-reference metrics try to avoid the practical problem with full-reference metrics by not using the undistorted reference in the comparison directly, but rather by using only features extracted form the reference as its reduced representation. As the data representing a number of feature values is usually much smaller than the amount of data necessary to represent the reference, it is feasible to provide these features as additional information with the distorted video. However, it must be ensured e.g. in a transmission system that this additional information is always available.

No-reference metrics represent the most general class of metrics in the context of reference availability. Except for the distorted video, no further information is required to estimate the video quality. Therefore no-reference metrics can be considered the most universal class of metrics for practical applications. The downside of this universality, however, is that no feature comparison of distorted and undistorted version of the video can be used in the quality estimation and thus the features must be able to describe distortion adequately, irrespective of the content.

3.3. Evaluation of the state-of-the-art

In this section, I briefly review and evaluate the state-of-the-art in video quality metrics. Each metric will be described briefly and assessed according to their compliance with the requirements defined in Section 3.1. The results of this assessment are noted in Table 3.2 to Table 3.8. I mainly review metrics based on the engineering approach, as this is the approach followed in the design of example video quality metrics in this thesis.

Additionally, I also include image quality metrics in this review, as on the one hand there are significantly more dedicated image quality metrics available, but on the other hand every image quality metric can also be applied to video frame-by-frame. In this context, it will be assumed that the overall visual quality for a complete video sequence is gained by applying the image quality metric separately to each frame, followed by averaging over the quality predictions of each frame in the video.

In the evaluation of the existing metrics, I use the information provided by the proponents of the assessed metrics in their original contribution for the assessment of the prediction performance and validation. It should be noted, however, that independent verification of metrics sometimes reveals that the prediction performance is significantly reduced for different images or videos not considered in the original contributions. An example for this issue is provided by Lin and Kuo [181], where the prediction performance for some image quality metrics fluctuates between $r_p = 0.6$ and $r_p = 0.95$, depending on the used images.

For further information about the metrics, I refer to the provided references, and for an additional review of the state-of-the-art I refer to the surveys by Chikkerur et al. [44], Lin and Kuo [181], Moorthy and Bovik [215] and You et al. [374].

3.3.1. Psychophysical-based

Even tough the focus in thesis is not on video quality metrics based on the HVS, I provide a short overview of psychophysical-based metrics for completeness. The *psychophysical* approach relies primarily on a (partial) model of the HVS and tries to exploit known psychophysical effects, e.g. masking effects, adaptation and contrast sensitivity. These effects are often depending on the spatial and temporal frequency of the input stimuli, and therefore the HVS is often modelled with a *multi-channel* characteristic, where each channel describes a spatio-temporal frequency range. If this channel dependency is omitted for simplicity, the resulting model is a simpler *single-channel* model. Note, that all models discussed in this section are full-reference metrics. For more information about psychophysical-based video quality metrics, I refer to Wu and Rao [368] and Winkler [354], and for a more fundamental discussion about human vision in general I therefore refer to Wandell [333].

Single-channel Faugeras [59] suggested the first HVS-based single-channel image quality models by using opponent-colour signals with their corresponding contrast sensitivity function and an approximation of the spatial sampling rate in the human eye. This was later extended by Lukas and Budrikis [191] to video sequences with a video quality metric considering spatio-temporal effects resulting in masking, followed by a Minkowski summation over all frames in order to arrive at an overall video quality. Tong et al. [322] extended this further in a video quality metric by using the CIE L*a*b* colour space.

Multi-channel In Daly's *visual differences predictor VDP* [48] image quality metric the adaptation 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. Lubin's *visual discrimination model VDM* [189] for images first convolves the input with an approximation of the eye's point spread function followed by an approximation of the eye's sampling characteristic and includes using a Laplacian pyramid for the spatio-temporal decomposition into different channels to apply directional filtering and masking.

The VDM was extended to video with the Sarnoff *just noticeable difference (JND)* [27, 190] video quality metric, where the pooling over different frames is performed by evaluating the 90% percentile of the individual frames' quality values. The *moving pictures quality metric (MPQM)* video quality metric by van den Branden Lambrecht and Verscheure [21] uses Gabor filters to approximate the spatio-temporal filter characteristics of the HVS into different channels and then applies contrast sensitivity functions and masking, followed by Minkowski summation over all frames. The *normalization fidelity metric (NVFM)* video quality metric by Lindh and van den Branden Lambrecht [183] extends the image quality model by Teo and Heeger [318] that uses a pyramid decomposition and a normalisation over all orientations, from images to video by adding a low delay temporal filter bank and

Table 3.2.: Compliance with the requirements in each category for metrics based on the human visual system (HVS)

Name	Compliance				Type
	Temporal	Reference	Prediction	Validation	
Single-channel					
Faugeras [59]	○○	○○	–	○○	IQ
Lukas and Budrikis [191]	○●	○○	○○	○○	VQ
Tong et al. [322]	○●	○○	–	–	VQ
Multi-channel					
VDP [48]	○○	○○	–	–	IQ
VDM [189]	○○	○○	–	–	IQ
Sarnoff JND [27, 190]	○●	○○	●●	○●	VQ
MPQM [21]	○●	○○	–	–	VQ
NVFM [183]	○●	○○	–	–	VQ
PDM [349, 353, 354]	○●	○○	○●	–	VQ
DVQ [343]	○●	○○	–	○●	VQ
Masry and Hemami [203]	●●	○○	–	○●	VQ

compliance level: ●●= full, ○●= acceptable, ○○= insufficient; – = no information available

also taking into account inter-channel masking effects, followed by Minkowski summation over all frames.

Winkler's video quality *perceptual distortion metric (PDM)* [349, 353, 354] includes a perceptual decomposition with contrast sensitivity and gain control, again followed by Minkowski summation over all channels and frames. The *Digital Video Quality (DVQ)* video quality metric by Watson et al. [343] metric uses a similar approach to the PDM, but uses the faster DCT instead of a filter bank to separate the different channels. Masry and Hemami [203] use a similar structure as above, but propose a continuous video quality metric by taking into account the quality of the preceding frames in the quality estimation of the current frame.

3.3.2. Pixel-based

Pixel-based metrics are often based on a common underlying principle that uses features exploiting certain properties of the HVS. If possible, metrics using the same principle are therefore grouped together. Similarly, metrics described in the same standard are also grouped together. Otherwise they are listed in one of the following three general categories: *full-reference*, *reduced-reference* or *no-reference* metrics.

PSNR derived metrics Although PSNR itself in not a suitable image and video quality metric as already discussed in Section 2.2, some modifications of the full-reference PSNR have been proposed that take into account properties of human perception: *PSNR-HVS* [52] determines the PSNR of a DCT version of the image weighted by a contrast sensitivity function and the *PSNR-HVS-M* [259] extends this by including an additional masking model.

Chandler and Hemami [40] use in the *visual signal to noise ratio (VSNR)* a discrete wavelet transform for a perceptual-like decomposition of the image and then assess the detectability of distortions in the resulting sub-bands, providing a measure of the perceived contrast of the distortions. The VSNR is then the PSNR between the contrast of the reference and perceived contrast of the distortions pooled over all sub-bands.

Wang and Li [340] suggest with the *information content weighted PSNR (IW-PSNR)* to apply a Laplacian pyramid transform, followed by weighing in each scale and for each coefficient the difference between reference and distorted image with a weight based on the mutual information between reference and distorted image.

Ou et al. [232] suggest the *VQMTQ* video quality metric for video with variable frame rate that uses a temporal correction factor (TCF) on the PSNR based on motion parameters, differences between consecutive frames and the amount of detail, represented by the strength and variance of the edges in the reference. The TCF not only adjusts for the perceptual difference in the frame rates, but also improves the prediction ability of the PSNR significantly.

Oelbaum et al. [229] suggested with $PSNR^+$ a correction step to increase the prediction performance of PSNR by generating two additional versions of the reference video on the opposite ends of the quality spectrum, encoding the reference with a high and low compression ratio. These high and low quality versions of the reference give an indication of the compressibility of the content represented by the reference and are then used to correct the PSNR, resulting in $PSNR^+$.

Visual information The full-reference *information fidelity criterion (IFC)* image quality metric by Sheikh et al. [288] uses a wavelet decomposition to decompose the image and then determines the mutual information between the reference and the distorted image, where the assumption is that natural i.e. undistorted images in the wavelet domain are normally distributed and the distortion process is similar to noise influencing a virtual channel over which the reference is transmitted. This approach was later extended by Sheikh and Bovik [287] in the full-reference *visual information fidelity (VIF)* image quality metric, where additionally the inherent *perceptual noise* that distorts both reference and distorted image is considered. The VIF is then defined as the ratio between the mutual information contained in the distorted image and reference, each with respect to the perceived, perceptual noisy stimulus.

Table 3.3.: Compliance with the requirements in each category for metrics based on PSNR, visual information and structural similarity

Name	Compliance				Type
	Temporal	Reference	Prediction	Validation	
PSNR derived metrics					
PSNR-HVS [52]	○○	○○	–	–	IQ
PSNR-HVS-M [259]	○○	○○	–	–	IQ
VSNR [40]	○○	○○	○●	○●	IQ
IW-PSNR [340]	○○	○○	●●	●●	IQ
VQMTQ [232]	○●	○○	●●	●●	VQ
PSNR$^+$ [229]	○○	○○	○○	○●	IQ
Visual information					
IFC [288]	○○	○○	●●	○●	IQ
VIF [287]	○○	○○	●●	○●	IQ
Structural Similarity					
SSIM [338]	○○	○○	●●	○●	IQ
MS-SSIM [334]	○○	○○	●●	○●	IQ
IW-SSIM [340]	○○	○○	●●	●●	IQ
VSSIM [341]	○●	○○	●●	○●	VQ
MC-SSIM [212]	○●	○○	○●	●●	VQ

compliance level: ●●= full, ○●= acceptable, ○○= insufficient; – = no information available

Structural similarity The *Structural SIMilarity index (SSIM)* is a full-reference image quality metric introduced by Wang et al. [338]. It calculates a quality index based on a comparison of the luminance, contrast and structure between sub-blocks of reference and distorted image, before pooling them for each image. Due to its comparably simple algorithm, the SSIM has become one of the currently most popular image and video quality metrics. Brunet et al. [34] have recently also proven that the SSIM has similar mathematical properties to the MSE.

The original SSIM has been modified by Wang et al. [334] in the *multi-scale SSIM (MS-SSIM)*, where the reference and distorted image are subsampled before applying the SSIM in each scale. The resulting SSIM value for each scale is then pooled into an overall MS-SSIM value for the complete image, leading to a further improvement in the prediction performance and Charrier et al. [41] provided a description of the MS-SSIM with optimised weighting parameters for the pooling over the different scales. Wang and Li [340] proposed a *information content weighted MS-SSIM (IW-SSIM)* that extends the MS-SSIM by including for each scale and block a weight based on the mutual information between reference and distorted image.

SSIM was extended to video in the *Video SSIM (VSSIM)* by Wang et al. [341]. For each frame the SSIM is determined similar to individual images, but in pooling over all frames by averaging the SSIM of each frame is weighed with a factor taking into account the motion between two consecutive frames. A different approach to extend SSIM to video was suggested by Moorthy and Bovik [212] in the *Motion-Compensated SSIM (MC-SSIM)*: for each frame in the reference and distorted video, each block is motion compensated with respect to the preceding frame, finding the corresponding block in the preceding frame. The SSIM is then determined between the same motion compensated block in the reference and the distorted video, followed by a percentile pooling over all blocks, before averaging over all frames, resulting in the overall quality rating of the video.

Natural scene statistics Metrics based on natural scene statistics (NSS) use the fact that *natural images* without distortion exhibit a certain distribution of the luminance [276]. Natural images in this context refers to realistic images as captured with a camera.

BLind Image Integrity Notator using DCT-Statistics (BLIINDS) is a no-reference image quality metric proposed by Saad et al. [278]. It uses the statistics of the distorted image's DCT coefficients to determine the image quality based on a probabilistic model of the relationship between statistical properties of the DCT coefficients and the subjective quality gained in the training with a set of images with disjunct content but similar distortions and known MOS. An improved version by the same authors, *BLIINDS-II* [279], fits a generalised gaussian model to the DCT coefficients and uses the shape parameters of this model, resulting in an improved prediction performance compared to BLIINDS. BLIINDS-II was also extended to video with the *Video-BLIINDS* by Saad and Bovik [277], where the DCT is performed on the difference between two consecutive frames and the parameters are determined on this difference frame. The pooling over all frames is then done by averaging.

Distortion Identification-based Image Verity and INtegrity (DIIVINE) by Moorthy and Bovik [213] is a similar no-reference image quality metric, but uses an overcomplete wavelet transform [294] to gain coefficients describing the image. Using these coefficients, DIIVINE then classifies the distortion into different categories e.g. JPEG or JPEG2000 and applies a distortion specific model for the quality prediction, obtained by a state vector regression on images with this distortion type. This approach was extended in the *Blind/Referenceless Image Spatial QUality Evaluator (BRISQUE)* by Mittal et al. [206], but instead of using coefficients in the transform domain, the statistical properties are directly gained from the spatial domain. Additionally, only one general model is used and no separate models for different distortion types are maintained. In [207], BRISQUE is also used for an unsupervised learning approach to image quality using probabilistic latent semantic analysis, but the prediction performance is even worse than PSNR.

Natural Image Quality Evaluator (NIQE) [208] is the latest no-reference image quality metric in this line. Unlike BLIINDS, DIIVINE and BRISQUE, NIQE is purely trained on undistorted reference images and without the use of corresponding MOS values. It par-

Table 3.4.: Compliance with the requirements in each category for metrics based on natural scene statistics and Gabor filter

Name	Compliance				Type
	Temporal	Reference	Prediction	Validation	
Natural scene statistics					
BLIINDS [278]	○○	●●	○○	○●	IQ
BLIINDS-II [279]	○○	●●	●●	●●	IQ
Video-BLIINDS [277]	○●	●●	○○	○●	VQ
DIIVINE [213]	○○	●●	●●	○●	IQ
BRISQUE Mittal et al. [206]	○○	●●	●●	○●	IQ
NIQE [208]	○○	●●	●●	○●	IQ
Gabor filter					
GDA [78]	○○	○○	○●	○●	VQ
RRP [78]	○●	●●	○○	○○	VQ
MAD [167]	○○	○○	●●	●●	IQ
ST-MAD [332]	●●	○○	○○	○●	VQ
MOVIE [282]	○●	○○	○●	○●	VQ

compliance level: ●●= full, ○●= acceptable, ○○= insufficient; – = no information available

titions the images in patches and then uses the same statistical features as in BRISQUE to describe the patches' properties, followed by fitting a multivariate Gaussian model. The quality is then expressed as the difference between the model gained with the training data and the model gained with the distorted image.

Gabor filter Gabor filter are often integrated in image and video quality metrics as they are a good representation of the properties of cells in the human visual cortex [194].

Guo et al. [78] use in their full-reference *Gabor Difference Analysis (GDA)* video quality metric multi-scale two dimensional Garbor filter on the frames of the reference and distorted video. For each frame, the difference between filtered reference and distorted image is then summed using Minkowski summation over all sub-bands, before pooling over all frames by averaging, resulting in an overall quality value. The authors extend this concept to the no-reference metric *Reverse Frame Prediction (RFP)* by assuming that the first frame in the complete video sequence or after a scene cut has relatively high quality and can therefore be considered as a reference, followed by applying the GDA between this pseudo-reference and all other frames.

The *Most Apparent Distortion (MAD)* full-reference image quality metric by Larson and Chandler [167] consists of a dual strategy, pooling the results from two different method:, one detection-based method using a perceptual weighted MSE for high quality images, and one appearance-based method for low quality images that uses a multi-scale two dimen-

sional Gabor filter to decompose the image into sub-bands, followed by a comparison of the sub-band statistics of the reference and distorted image. Vu et al. [332] extended MAD to the *spatiotemporal MAD (ST-MAD)* video quality metric. It combines the MAD averaged over all frames with a motion-based distortion value that is estimated using spatio-temporal slices in combination with motion weights for each row and column of the slices.

The *MOtion-based Video Integrity Evaluation index(MOVIE)* full-reference video quality metric by Seshadrinathan and Bovik [282] uses a multi-scale three dimensional Gabor filter to decompose the reference and distorted video into different sub-bands, each representing a certain spatio-temporal band and orientation. The maximum contrast normalised difference between reference and distorted video in each sub-band results in the spatial component of the quality index, whereas the temporal component represents the deviation in motion between reference and distorted video. Both components are then averaged over all frames and combined by multiplication.

Data analysis Shnayderman et al. [291] presented a full-reference grey-scale image quality metric based on a block-wise SVD of both the reference and distorted image, followed by calculating the MSE between the singular values for each block. The overall quality value is then determined by the MSE between the MSE of each block and the median MSE of all blocks.

Narwaria and Lin [218] also use a SVD in their full-reference image quality metric, but combines it with a support vector regression on MOS scores, leading to better prediction results.

Cheng and Wang [43] extend the SVD approach to the full-reference colour image quality metric *M-TDc* by considering each colour image as a three-way array and then applying a Tucker3 decomposition as a higher order equivalent to the SVD on three-way blocks of the image. The overall quality value is then determined by the weighted sum of the correlation coefficients between the components of the reference and distorted image, where the weights were determined by training.

Miyahara [209] first suggested a full-reference image quality metric based on a principal component regression (PCR) between features and MOS with the *Picture Quality Scale (PQS)*, where the features were bases on perceptually weighted differences between reference and distorted image. This was extended in an improved version of the PQS in [210] by also including blocking and accounting for visual masking effects.

Oelbaum and Diepold [228] propose a reduced reference video quality metric based on partial least squares regression (PLSR) that includes both spatial distortion and frame difference features that are extracted from each frame and then averaged over all frames for both reference, distorted video and a low quality version of the reference generated by encoding the reference with a high compression ratio, similar to the approach suggest in [229]. The quality prediction of the distorted reference is then gained by a weighed sum of the features, corrected with a quality prediction of the reference and the low quality version of the reference using the same prediction model in order to account for the

Table 3.5.: Compliance with the requirements in each category for metrics based on data analysis

Name	Compliance				Type
	Temporal	Reference	Prediction	Validation	
Data analysis					
Shnayderman et al. [291]	○○	○○	●●	○○	IQ
Narwaria and Lin [218]	○○	○○	○●	●●	IQ
M-TDc [43]	○○	○○	●●	○●	IQ
PQS [209]	○○	○○	–	○○	IQ
PQS [210]	○○	○○	–	○●	IQ
Oelbaum and Diepold [228]	○●	○●	○○	○●	VQ
Oelbaum and Diepold [227]	○●	○●	○●	○●	VQ
Oelbaum et al. [224]	○●	●●	○○	○●	VQ

compliance level: ●●= full, ○●= acceptable, ○○= insufficient; – = no information available

compressibility of the reference's content. The weights are determined in a training by building a prediction model between features and corresponding MOSs with PLSR. This was slightly extended by the authors in [227] by including a non-linear post processing step to account for the contraction bias often encountered in the results of subjective testing.

Oelbaum et al. [224] used a similar approach as in [227] to design a no-reference video quality metric. Although the same features are used as in [227], the correction step has been modified so that only a low quality version of the distorted video is necessary and thus no information about the reference needs to be provided. In addition, depending on the extracted features, different prediction models are selected, each optimised for a different distortion category.

Full-reference metrics You et al. [372] suggest a video quality metric that takes into account a visual attention model. For each frame, the squared error of the luminance and chrominance between reference and distorted video, and the ratios between reference and distorted video with respect to inter frame continuity between neighbouring frames, horizontal, vertical and diagonal edge preservation is determined both for the complete frame and separately for the most attention receiving regions in the frame. These features are then pooled by averaging over all frames and weighed with a measure of the spatio-temporal complexity for the attention receiving regions, resulting in the overall quality.

Ninassi et al. [220] propose a method with spatio-temporal tubes that span multiple frames of the video, resulting in a spatio-temporal distortion map for each frame. Hence, the influence of temporal variation on areas with high attention is weighed differently than on those with less attention. The resulting distortion maps per frame is corrected with a term representing the temporal gradient of the per frame distortion before being averaged over all frames, resulting in an overall quality value.

Barkowsky et al. [11] suggest within the *TetraVQM* framework a video quality metric that uses spatial, frame-wise error estimation with PSNR or SSIM weighted with the temporal visibility duration of each frame's region derived over all previous frames, emulating the tracking of objects by the test subjects in subjective testing. This results in representative quality value per frame that are then average of all frames and combined with information about frame freezes and skips, resulting in an overall quality value.

Reduced-reference metrics The *Reduced Reference Entropic Differencing (RRED)* image quality metrics by Soundararajan and Bovik [302] uses the difference in the entropy of the coefficients of a wavelet transformation between the reference and distorted images as a quality measure, where the reference coefficients' entropy represents the reduced reference information. This was extended by the authors to a video quality metric in [303] by including additionally the entropy of the wavelet representation of the difference between two consecutive frames, followed by an averaging over all frames.

Redi et al. [265] propose a image quality metric using statistical information about the colour distribution in the reference as reduced reference information. It uses a combination of a support vector machine to classify the distortion and a circular back propagation neural network to provide the quality estimation, where for each distortion type a separate classifiers and estimator exists.

Le Callet et al. [172] present a video quality metric that uses frame differences classified according to their severity, blurring and blocking as features. These features represent the reduced reference information and are extracted from both reference and distorted video. The features are then fed into a time delay neural network that takes into account the preceding 5 s of the video for each frame. The neural network has been trained on MOSs from a continuous quality evaluation with the SSCQE method, thus allowing the calibration of the network for a continuous quality prediction. However, no overall pooling into a single quality value is provided.

No-reference metrics Farias et al. [57] in their video quality metric firstly use a DCT on each frame of the reference, followed by embedding a visually invisible watermark into the DCT coeffcients. An inverse DCT is then applied and the resulting image with embedded watermark is processed as usual, resulting in a distorted image. To assess the quality, a DCT is performed on the frames of the distorted video and the watermark extracted from the coefficients. The MSE between the watermark of the distorted video and the original watermark calculated over all frames is then used as an indicator of the visual quality. Note, that one could also classify this metric as a reduced-reference metric, but Farias et al. argue that no additional data needs to be transmitted and the watermark embedding can be part of the overall system.

Farias and Mitra [58] propose a video quality metric that determines blockiness, blurring and noisiness for each frame, followed by averaging over all frames. The features are differently weighed and combined with Minkowski summation.

Table 3.6.: Compliance with the requirements in each category for various full-reference, reduced-reference and no-reference metrics

Name	Compliance				Type
	Temporal	Reference	Prediction	Validation	
Full-reference					
You et al. [373]	○●	○○	○●	●●	VQ
Ninassi et al. [220]	●●	○○	○●	○●	VQ
TetraVQM-SSIM [11]	●●	○○	○○	○●	VQ
Reduced-reference					
RRED [302]	○○	○●	○●	●●	IQ
Video-RRED [303]	○●	○●	○○	○●	VQ
Redi et al. [265]	○○	○●	○●	○●	IQ
Le Callet et al. [172]	●●	○●	●●	○○	VQ
No-reference					
Farias et al. [57]	○○	●●	○●	○○	VQ
Farias and Mitra [58]	○●	●●	○●	○○	VQ
Yang et al. [370]	○●	●●	○○	○●	VQ
Chen and Bovik [42]	○○	●●	–	○●	IQ
Kawayokeita and Horita [135]	●●	●●	○●	○○	VQ
Gastaldo et al. [70]	○○	●●	●●	○●	IQ

compliance level: ●●= full, ○●= acceptable, ○○= insufficient; – = no information available

Yang et al. [370] identify in their video quality metric regions with high spatial complexity by comparing the consistency of motion vectors in neighbouring blocks. Then the difference between the same region in the current and previous frame is determined both for the unprocessed frame and a smoothed version of the frame, resulting in the overall spatial distortion of the frame that is then weighted with the average motion vector length in the frame to account for the temporal activity, resulting in the distortion of the frame. The overall quality is then determined by averaging over all frames.

Chen and Bovik [42] use in their image quality metric a state vector machine (SVM) to classify images either as blurred or sharp, and depending on the classification and confidence of the classification assign define a different so-called coarse quality score. Using a wavelet decomposition of the distorted images, a detail score based on the gradient is determined in each sub-band. Depending on the sub-band, the detail score is then assigned a different factor in the exponent and the overall blur value is calculated as the product of all sub-bands' detail scores and the coarse quality score.

Kawayokeita and Horita [135] propose to use histograms for each frame describing the difference between the distorted video and a version of the distorted video filtered with an edge preserving filter. The histograms are then summed up for each frame, followed by an averaging for all frames in a 0.5 s interval of the video, representing the quality of this interval. Additionally, in a separate training the sum for each frame is used together with the spatial and temporal information from the distorted frame to determine a correction factor depending on these parameters. Following the application of the correction factor, the quality for each 0.5 s interval is then mean filtered to remove high frequency fluctuations in the prediction, resulting finally in a continuous quality prediction. However, no overall pooling into a single quality value is provided.

Gastaldo et al. [70] suggest a no-reference image quality metric using six features derived from the statistics of colour correlograms. Colour correlograms express how the spatial correlation of colour pairs change with spatial distance. It then trains a circular back propagation neural network with these features and corresponding MOSs, resulting in prediction weights for predicting the quality.

Standardised full-reference video quality metrics One practical feature usually encountered in standardised full-reference video quality metrics metrics is the inclusion of spatial and temporal alignment mechanisms, as in real-life scenarios it can not necessarily be assumed that reference and distorted video are synchronous to each other.

ITU-T J.144 [116] suggests a set of four different video quality metrics: the first metric in ITU-T J.144, Annex A or *BTFR* metric uses a combination of matched PSNR, PSNR of a pyramidal decomposition, edge differences and a texture analysis based on turning points for each frame, followed by an averaging over all frames. ITU-T J.144, Annex B or the *Yonsei* metric uses edge detection in each frame, followed by the calculation of the PSNR of the overall difference between the edges in the reference and distorted video in the complete video, resulting in the overall quality that is additionally corrected for the blur encountered in the distorted video. Therefore this method is sometimes also referred to as *EdgePSNR*. ITU-T J.144, Annex C or the *CPqD* metric segments first the reference video into edges, planes and texture, and additionally generates two degraded version of the reference video by encoding with a MPEG-2 and MPEG-1 encoder. For each of the three versions, the difference between the Sobel filtered representation of the reference and distorted video is calculated in each frame and colour channel, and taking into account the corresponding segment this results in a objective measures for each frame. These objective measures are then used together with an impairment database to assign an impairment level to each frame, before averaging over all frames. ITU-T J.144, Annex D or the *NTIA* metric [366, 367], calculates spatial and temporal short-term luminance gradients, representing the contrast, motion and activity, for both reference and distorted video. The overall quality value is achieved by temporal pooling using 10% and 90% percentiles that are representative of the worst transient quality for gains and losses in

the video sequence. Depending on the context, different prediction models for different applications can be chosen. Note, that ITU-R BT.1683 [103] is equivalent to ITU-T J.144.

ITU-T J.247 [119] suggests a set of four different video quality metrics: the first metric in ITU-T J.247, Annex A or the *NTT* metric uses the average PSNR over all frames, the minimum blockiness encountered in the distorted video, the moving energy of the blocks as indicator for motion distortion and frame freeze length. Each feature is pooled over all frames and then combined in an overall quality estimation. ITU-T J.247, Annex B or the *Opticom PEVQ* metric first defines a a region of interest (ROI). Then the edgeiness of reference and distorted video for each frame is determined, followed by assessing the visibility of the edges with respect to the luminance and chrominance. Additionally, the frame repetition and frame freeze length are determined for each frame. Each feature is pooled over all frames and then combined in an overall quality estimation. ITU-T J.247, Annex C or the *Psytechnics* metric is similar to the structure of the BTFR, but determines for each frame also blocking and blurring, complemented by an indicator for frame freezes. Each feature is pooled over all frames and then combined in an overall quality estimation. ITU-T J.247, Annex D is similar to ITU-T J.144, Annex B. It determines the EdgePSNR without correction for blur but with a correction for frame freezes. Additionally, the amount of blocking and blurring is determined over the complete video sequence. All there features are then combined in an overall quality estimation. Note, that ITU-R BT.1866 [105] is equivalent to ITU-T J.247.

ITU-T J.341 [124] is aimed at HDTV and uses local similarity and the difference between reference and distorted video in each frame combined with a jerkiness detection for each frame. The overall quality is then gained by averaging over all frames.

ITU-R BT.1907 [111] is equivalent to ITU-T J.341 except that additionally the blockiness of the distorted video is included in the spatial features besides local similarity and also the difference between reference and distorted video.

Standardised reduced-reference video quality metrics ITU-T J.249 [123] suggests a set of three different video quality metrics: the first metric in ITU-T J.249, Annex A is a variation of ITU-T J.144, Annex B: edges are detected in a window spanning multiple frames in both the reference and the distorted video. The edges found in the reference represent the reduced reference information and after spatial and temporal registration the corresponding edges found in the distorted video are used to determine the MSE between edges in the reference and distorted video. The overall quality value is then determined by averaging the MSE over all values adjusted by a weights taking into account freezed frames, fast motion, blurring and blocking. ITU-T J.249, Annex B or the *NEC* metric determines an activity value for the corresponding 16×16 blocks in both reference and distorted video, representing the reduced reference information. The square error between these activity values for each block is then weighed in order to take spatial frequency, colour and scene changes into account. Then the PSNR of the average squared error over of all blocks and frames is determined, followed by weighing of the PSNR with a factor

Table 3.7.: Compliance with the requirements in each category for standardised full- and reduced-reference metrics

Name	Compliance				Type
	Temporal	Reference	Prediction	Validation	
Full-reference					
ITU-T J.144, Annex A [116]	○○	○○	○○	○●	VQ
ITU-T J.144, Annex B [116]	○●	○○	○●	○●	VQ
ITU-T J.144, Annex C [116]	○○	○○	○●	○●	VQ
ITU-T J.144, Annex D [116]	○●	○○	●●	●●	VQ
ITU-T J.247, Annex A [119]	○○	○○	○○	●●	VQ
ITU-T J.247, Annex B [119]	○●	○○	○○	●●	VQ
ITU-T J.247, Annex C [119]	○●	○○	○○	●●	VQ
ITU-T J.247, Annex D [119]	○●	○○	○○	●●	VQ
ITU-T J.341 [124]	○●	○○	○●	●●	VQ
ITU-R BT.1907 [111]	○●	○○	○●	●●	VQ
ITU-R BT.1908 [112]	○●	○○	○○	●●	VQ
Reduced-reference					
ITU-T J.246 [120]	○●	○●	○○	●●	VQ
ITU-T J.249, Annex A [123]	○●	○●	●●	○●	VQ
ITU-T J.249, Annex B [123]	○○	○●	○○	○●	VQ
ITU-T J.249, Annex C [123]	○●	○●	○●	●●	VQ
ITU-T J.342 [125]	○●	○●	○○	●●	VQ

compliance level: ●●= full, ○●= acceptable, ○○= insufficient; – = no information available

for average blockiness and impairment, resulting in the final quality. ITU-T J.249, Annex C represents a derivation of the NTIA metric with a focus on providing low bandwidth reduced reference information. Therefore only a subset of all in the NTIA metric available features are used in order to minimise the overhead due to the reduced reference information. The individual features are pooled temporally with Minkowski summation.

ITU-T J.246 [120] is similar to ITU-T J.144, Annex B, but only a correction for frame freezes is applied on the computed EdgePSNR. Note, that ITU-R BT.1867 [106] is equivalent to ITU-T J.246.

ITU-T J.342 [125] aimed at HDTV is similar to ITU-T J.249, Annex A, but the correction term only represents the average blocking encountered in the distorted video.

ITU-R BT.1908 [112] is similar to ITU-T J.249, Annex A, but the correction term only represents the average blocking encountered in the distorted video and freezed frames. Additionally, also a correction for transmission errors is introduced into the overall quality value.

Pooling for image quality metrics Although the assumption is made in this review that image quality metrics are applied frame-by-frame, followed by an averaging over all frames, other pooling option are possible in order to extend image quality metrics to video.

Rimac-Drlje et al. [270] compared different pooling methods for different image quality metrics and suggest Minkowski summation over all frames, but results by You et al. [372, 374] indicate that Minkowski summation does not provide any advantage compared to simple averaging.

Park et al. [237, 238] proposed a modification of simple averaging for frame-based quality metrics by first assigning each frame to a group with high or low quality based on the results of the frame-based metric for each frame and k-means clustering, then defining weights for each group based on the average quality in this group, followed by averaging over the weighted frames. Alternatively, Lee et al. [177] suggests to use the i-th percentile of the lowest quality and its compliment to assign the frames to different quality groups.

3.3.3. Bitstream-based

Bitstream-based metrics are specifically defined for certain coding technologies and are therefore grouped according to coding technology in this section.

DCT coefficients Bitstream-based in this context address all coding technologies that utilise the DCT.

Watson [342] proposed the full-reference image quality metric *DCTune* that uses the spatial-frequency decomposition of the DCT and weighs the difference in the DCT coefficients between reference and distorted image with luminance and contrast masking.

Brandão and Queluz [22] suggest a no-reference image quality metric using an estimation of the coefficients in the DCT representation of the reference, based only on the quantised coefficients in the distorted image. In order to achieve this, the parameters of the DCT coefficients distribution in the reference are estimated by taking into account the correlations between the neighbouring frequencies in the distorted image's DCT decomposition. It then uses this distribution to estimate the absolute error for each coefficient between its original and quantised version. After weighing the error depending on a just noticeable difference threshold, the weighed errors are pooled over all blocks using a modified L_4 norm.

H.264/AVC Because the focus in this thesis is on the quality prediction of distorted video caused by encoding and not by transmission errors, H.264/AVC bitstream-based metrics that utilise error statistics in their predition model e.g. [7, 8, 217, 263, 306, 307, 369, 370] are not considered in this section. Similarly, as PSNR itself can not be considered a quality metric, methods estimating the PSNR from the bitstream e.g. [50, 216, 297] are also not included.

Table 3.8.: Compliance with the requirements in each category for bitstream-based metrics

Name	Compliance				Type
	Temporal	Reference	Prediction	Validation	
DCT coefficients					
DCTune [342]	○○	○○	–	–	IQ
Brandão and Queluz [22]	○○	●●	●●	○●	IQ
H.264/AVC					
Brandão et al. [25]	○●	●●	●●	○●	VQ
Brandão and Queluz [23]	○●	●●	●●	○●	VQ
Brandão and Queluz [24]	○●	●●	●●	○●	VQ
Sugimoto et al. [314]	○●	●●	●●	○●	VQ
Sugimoto and Naito [313]	○●	●●	●●	○●	VQ
Lee et al. [178]	○●	●●	○●	○●	VQ

compliance level: ●●= full, ○●= acceptable, ○○= insufficient; – = no information available

Brandão et al. [25] propose a no-reference metric that uses the bit rate, an error estimation based on quantisation noise of the distorted video and spatial and temporal activity as suggested in ITU-T P.910 and the activities' variances, where the last three features require the decoding of the bitstream. The features are then combined in the quality prediction using weights determined in a training.

Brandão and Queluz [23] suggest a no-reference metric that uses for each frame a block-wise error estimation based on the quantisation noise of the distorted video, weighed with a spatio-temporal contrast sensitivity function derived from the frame rate and motion vector directions. The overall quality is then gained by applying the L_4 norm over all frames. This approach was extended by the authors in [24], using a coefficient distribution model for the error estimation of the quantisation noise and an additional weighing of the distributions parameter in order to take the correlations between neighbouring coefficients into account, where the weights are determined in a training.

Sugimoto et al. [314] proposes a hybrid no-reference metric combining bitstream-based feature with pixel-based features from the decoded bitstream. For the pixel-based features, the metric uses the average blockiness and flickering determined over all frames in the sequence, and as a bitstream-based feature it uses the average quantiser scale determined over all frames in the sequence. These three features are then combined in a overall qualiy value with Minkowski summation. This approach was refined to a pure bitstream-based metric by Sugimoto and Naito [313], where the blockiness is now determined with the difference of the transform coefficients between neighbouring macroblocks and flickering is approximated by the power of a macroblock's coefficients divided by the interval to the macroblock's reference frame, where the fact is used that the coeffienctes

represent the residual of the not modelled motion in relation to the macroblock's reference frame.

Lee et al. [178] use bitrate, quantisation point and the properties of macroblocks in their no-reference metric. Based on the macroblocks' properties and locations, a boundary strength is determined and classified into four categories. The overall quality is then determine by combining the average bitrate and quantisation points over all frames with the frequency of occurrence of the different boundary strength categories in a linear model.

3.3.4. Summary of the state-of-the-art

The evaluation of the state-of-the-art with respect to the requirements in this thesis can be summarised as following:

Temporal pooling Most metrics reviewed are image quality metrics and therefore due to the assumption of averaging for temporal pooling of the individual frames' do not consider the temporal dimension sufficiently.

Considering only quality metrics designed specifically as video quality metrics, most metrics account for the temporal nature of video by using more sophisticated pooling methods as Minkowski summation [21, 58, 183, 343, 349, 353, 354] or percentile pooling [27, 116, 190]. Some also include frames in a small interval around the current frame into the prediction [21, 183, 282] or use the differences to the preceding frame in the prediction [78, 212, 224, 227, 228, 232, 277, 341, 370, 373].

Only a small subset of all reviewed metrics consider the temporal dimension of video without any significant temporal pooling or at least consider relatively large interval of a few seconds [11, 135, 172, 203, 220, 332], but two of these metrics aim at providing continuous quality predictions and therefore provide no overall quality prediction [172, 203].

Reference availability Although we can observe a trend to no-reference metrics in image quality metrics e.g. [206, 208, 213, 278, 279], the majority of image quality metrics are still full-reference or reduced-reference metrics.

Similarly, most pixel-based video quality metrics are still full-reference or reduced-reference metrics and only a small subset consists of no-reference metrics exits [57, 58, 70, 135, 224, 370]. Note that there is no standardised pixel-based no-reference video quality metrics available, yet. In contrast, nearly all bitstream-based video quality metrics are no-reference metrics [23–25, 178, 313, 314].

Prediction performance Image quality metrics often achieve a high prediction performance with $r_P \geq 0.90$, contrary to the pixel-based video quality metrics, where only a small number reach this goal [27, 70, 172, 190, 232, 277, 341]

Bitstream-based video quality metrics are less universal, but due to the focus on a specific technology nearly all of these metrics are able to achieve a prediction performance of $r_P \geq 0.90$ [23–25, 313, 314].

Validation Validation is mostly limited to a single data set and only a small number of quality metrics are validated with multiple datasets [218, 232, 279, 302, 340, 373]. One advantage of standardised video quality metrics in this context is their comprehensive validation during the standardisation process.

Overall assessment Considering all the discussed metrics, we can notice that even tough many metrics fulfil two or even more of the criteria extremely well, no metric fulfils all our four requirements completely. In particular, even though the requirements with respect to the prediction performance and reference availability are met by both natural-scene-statistics-based image quality metrics [206, 208, 213, 278, 279] and bitstream-based video quality metrics [23–25, 313, 314], either some form of temporal pooling is applied and/or the validation is not comprehensive enough.

Part II.

Data Analysis

4. Data Analysis Approach

During the design of video quality metrics often very specific relationships between the objectively measurable features and the subjectively perceived quality are assumed. In doing so, however, two problems can arise: on the one hand the assumptions about the influence of the chosen features may be wrong, but on the other hand possible features may also be excluded beforehand even tough they provide in fact a significant contribution to the video quality. The danger is that this may not even be obvious, as there is no need to look beyond the predefined model based on the a-priori assumptions chosen before the design process started. Potential improvements based on either the excluded features or on an adjustment of the included features' influence might therefore neither be noticed, nor implemented. This leads to video quality metrics that are not utilising all available information for the quality prediction.

Data analysis provides a different approach. Instead of using any assumptions, we analyse the data without bias in a purely data driven approach. In the context of video quality metrics, the data consists of the objective features and subjective quality of the video sequences. The interactions between features and quality emerge during the analysis and are then used to build prediction models in order to estimate the subjective visual quality from the objective features.

In this chapter, I present the data analysis approach to model building and discuss how it can be used to design video quality metrics. After a general introduction into the data analysis approach and its application to the design of video quality metrics, I briefly introduce the used notation and the necessary preprocessing of the data. This chapter and the following chapters in this part of the thesis are mainly based on the contributions by Martens and Næs [197], Smilde et al. [298] and Bro [28].

4.1. Data analysis in the design of video quality metrics

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 analysed. 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 and a good introduction into chemometrics can be found in Martens and Martens [195].

Applying data analysis to video quality, we consider the HVS as a black box and therefore do not assume a complete understanding of it. The input corresponds to features representing properties of the video that can be measured objectively, and the output of the box to the perceived visual quality obtained in subjective tests. While this is similar to the engineering approach described in the previous chapter, 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. 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.

The data analysis approach can therefore be considered as *soft modelling* [197]: the aim is to understand the system, in our case the relationship between video sequences' properties and visual quality within the HVS, by observing the interdependencies that emerge during the analysis. Hence, this approach is *open-ended*, as unexpected phenomena that emerge during the analysis can be considered in the design of the model. This leads to a pragmatic model that describes the interdependencies of the data as well as possible, without the need for a complete causal model of the system beforehand. In contrast, the *hard modelling* starts from a well defined, but possibly incomplete causal model. Especially if the overall system, as in our case the HVS, is not completely understood, this can be dangerous as it *looks the mind into possibly inadequate explanations* [197]. One could argue, that data fitting, which is often used on the prediction results of video quality metrics in performance comparisons, is a consequence of the hard modelling: the model does not predict the data well enough, but instead of revisiting the assumptions beneath the model, the prediction results are fitted to the current data in order to improve the prediction performance of the video quality metric. For a more detailed discussion about the differences and benefits of soft and hard modelling, I follow the suggestion by Martens and Næs [197] and refer to Box et al. [19].

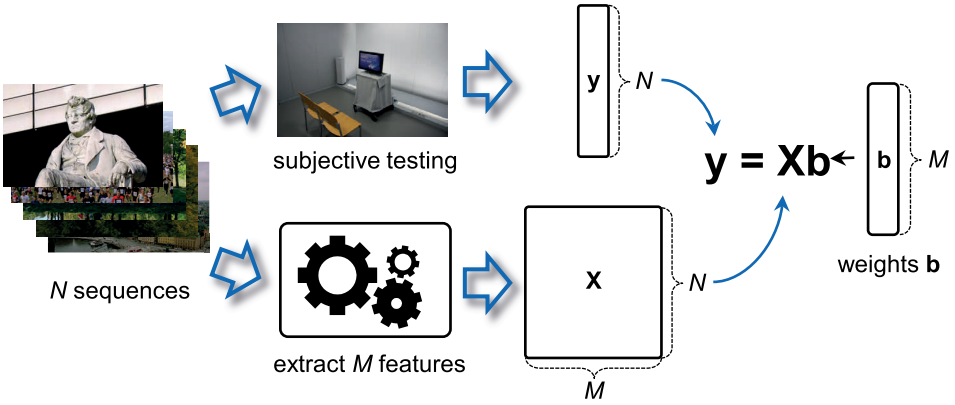

Figure 4.1.: Overview of model building with the data analysis approach: subjective testing and feature extraction for each video sequence

Describing the data analysis approach more formally and neglecting the temporal nature of video for simplicity, we firstly extract M features for each the N different video sequences, resulting in a $1 \times M$ row vector $\mathbf{x}$. Thus we arrive at an $N \times M$ matrix $\mathbf{X}$, where each row describes a different sequence or sample and each column describes a different feature. Secondly, 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 or described more formally, we want to solve the equation

$$\mathbf{y} = \mathbf{Xb} \tag{4.1.1}$$

for b. The basic model building or fitting process is illustrated in Fig. 4.1. This process is called *calibration* or *training* of the model, and the used sequences are the calibration or training set. We can then 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. The prediction performance is usually *validated* with sequences not used in the calibration, but with known visual quality. The calibration itself can be performed with different data analysis methods and the methods used in this thesis are discussed in Chapter 5 and Chapter 6.

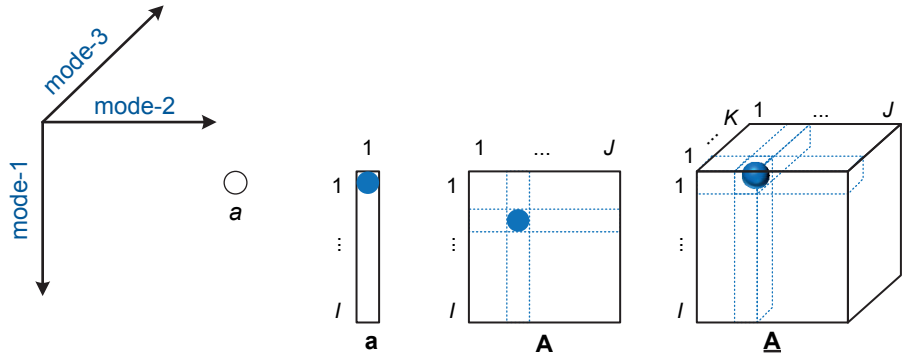

Figure 4.2.: Notation used in this thesis for scalars and different n-way arrays: scalars or zero-way arrays, vectors or one-way arrays, matrices or two-way arrays and three-way arrays (from left to right). The circle denotes an individual element of each array (derived from [298])

4.2. Preliminaries

This section discusses two issues that need to be addressed before the data analysis itself can be performed. Firstly, I discuss the notation that was briefly introduced in the previous section in detail. The aim is to provide a notation, that allows the unambiguous description of video sequences and their corresponding features. Secondly, the necessary preprocessing of the data is discussed.

4.2.1. Notation

In this thesis, the following notation is used: scalars are denoted as lower-case italic letters, e.g. a, vectors as lower-case bold letters, e.g. $\mathbf{a}$, matrices as upper-case bold letters, e.g. $\mathbf{A}$, and *multi-way arrays* as underlined upper-case bold letters, e.g. $\underline{\mathbf{A}}$. Estimations of variables are denoted by a circumflex above the variable, e.g. $\hat{a}$ denotes a as estimated by a model. Multi-way arrays, often also called *tensors*, can be considered as extensions of the matrix concept into a higher dimensional space: whereas a matrix represents a two-dimensional space, a multi-way array represents an n-dimensional space, with $n > 2$, e.g. a three-way array with $n = 3$ is representing a three dimensional space in the form of a cube or rectangular cuboid. Using the concept of multi-way arrays more in general as n-way arrays, we can also consider matrices as *two-way arrays*, vectors as *one-way arrays* and scalars as *zero-way arrays* for $n = 2$, $n = 1$ and $n = 0$, respectively.

The different ways are called *modes*, e.g. a three-way array has three modes, mode-1 to mode-3, corresponding to its three indices. Fig. 4.2 illustrates the used notation and the concept of multi-way arrays for the three-way case: shown are a scalar $a \in \mathbb{R}$, a vector $\mathbf{a} \in \mathbb{R}^I$, a two-way array $\mathbf{A} \in \mathbb{R}^{I \times J}$ and a three-way array $\underline{\mathbf{A}} \in \mathbb{R}^{I \times J \times K}$, where I, J, K

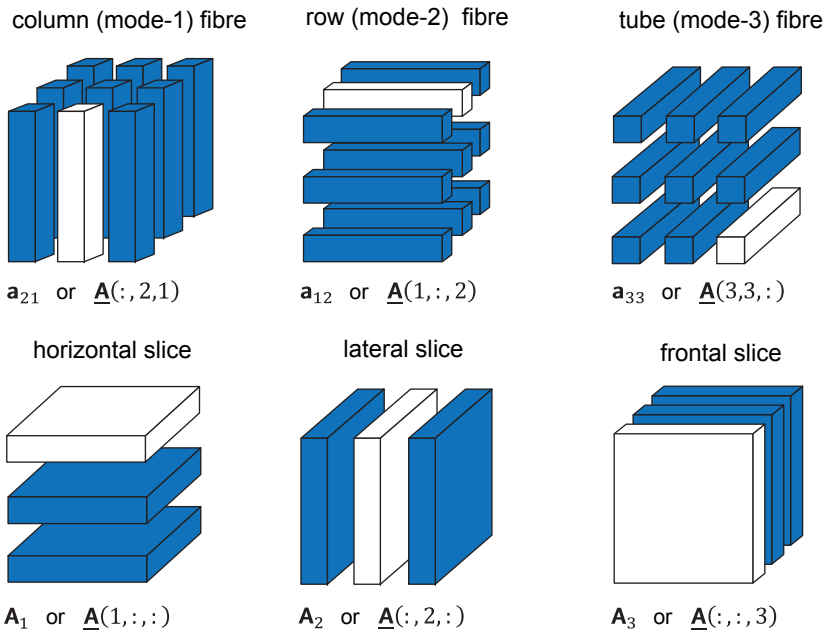

Figure 4.3.: Fibres and slices of a three-way array: examples for a three-way array $\underline{\mathbf{A}} \in \mathbb{R}^{3\times3\times3}$ (derived from [45])

denote the dimensionality of each mode and it is assumed without loss of generality that the elements of the n-way arrays are real-valued. For two-way arrays, row and column vectors are differentiated by their definition as $\mathbf{a} \in \mathbb{R}^{1\times J}$ and $\mathbf{a} \in \mathbb{R}^{I\times1}$, respectively. The individual elements of the n-way arrays shown in Fig. 4.2 on the preceding page can be expressed for one-way and two-way arrays as following: for a column vector $\mathbf{a} \in \mathbb{R}^{I\times1}$, a_i denotes the i-th entry in $\mathbf{a}$, where $i \in \mathbb{N}$ and $i = 1, \ldots, I$. Similarly, for a two-way array $\mathbf{A} \in \mathbb{R}^{I\times J}$, a_{ij} denotes the (i,j)-th element of $\mathbf{A}$, where $i = 1, \ldots, I$ and $j = 1, \ldots, J$ with $i, j \in \mathbb{N}$. Additionally, $\mathbf{a}_i$ and $\mathbf{a}_j$ denote the i-th row and j-th column of $\mathbf{A}$, respectively.

In order to address the elements of multi-way arrays appropriately, the concepts of rows and columns need to be generalized to *n-mode fibres*, where each fibre is a one-way array. As illustrated in Fig. 4.3 for three-way arrays, mode-1 and mode-2 fibres correspond to columns and rows in a matrix. Additionally, the three-way array has mode-3 fibres or *tubes*, that represent vectors in the direction of the third mode. Often it is useful to segment a multi-way array into two-way arrays in order to apply methods and algorithms for matrices to multi-way data. Therefore the one-way array fibre concept is extended to the two-way array with the *slice* concept as shown in Fig. 4.3 for three-way arrays. We can distinguish between mode-1 or *horizontal* slices, mode-2 or *lateral* slices and mode-3 or *frontal* slices.

Similar to the notation for two-way arrays, a_{ijk} denotes the (i,j,k)-th element in a three-way array $\underline{\mathbf{A}} \in \mathbb{R}^{I\times J\times K}$, where $i = 1, \ldots, I$, $j = 1, \ldots, J$ and $k = 1, \ldots, K$, with $i, j, k \in \mathbb{N}$. Considering the different fibres in $\underline{\mathbf{A}}$, $\mathbf{a}_{jk}$ denotes the (j,k)-th column with $\mathbf{a}_{jk} \in \mathbb{R}^I$, $\mathbf{a}_{ik}$

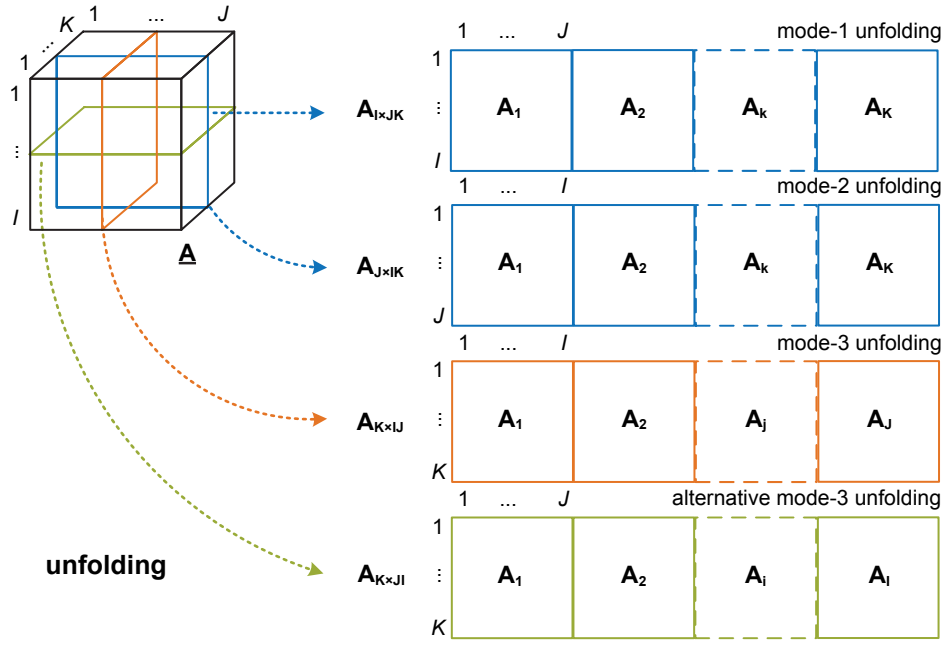

Figure 4.4.: Unfolding of a three-way array $\underline{\mathbf{A}} \in \mathbb{R}^{I \times J \times K}$ into a two-way array by concatenating frontal, lateral and horizontal slices (derived from [298])

the (i,k)-th row with $\mathbf{a}_{ik} \in \mathbb{R}^{J}$, and $\mathbf{a}_{ij}$ denotes the (i,j)-th tube with $\mathbf{a}_{ij} \in \mathbb{R}^{K}$. The i-th horizontal slice of $\underline{\mathbf{A}}$ is denoted as $\mathbf{A}_i$ with $\mathbf{A}_i \in \mathbb{R}^{J \times K}$, the j-th lateral slice as $\mathbf{A}_j$ with $\mathbf{A}_j \in \mathbb{R}^{I \times K}$, and the k-th frontal slice as $\mathbf{A}_k$ with $\mathbf{A}_k \in \mathbb{R}^{I \times J}$. Possible ambiguities are avoided by providing the context in the text where necessary.

Slices can be used to *matricise*, *flatten* or *unfold* the three-way array $\underline{\mathbf{A}}$ into a two-way array by concatenating the slices of $\underline{\mathbf{A}}$ into one matrix $\mathbf{A}$. In general, there are multiple ways to arrange the slices into a two-way array and the order of the fibres of $\underline{\mathbf{A}}$ in the unfolded matrix $\mathbf{A}$ is not unique. Following the definition proposed by Kolda and Bader [155], in this thesis the mode-n unfolding of $\underline{\mathbf{A}}$ is defined as the rearrangement of the mode-n fibres of $\underline{\mathbf{A}}$ into the columns of the unfolded matrix $\mathbf{A}$. Hence, the mode-1 unfolding of $\underline{\mathbf{A}}$ leads to $\mathbf{A}_{I \times JK} \in \mathbb{R}^{I \times JK}$, the mode-2 unfolding to $\mathbf{A}_{J \times IK} \in \mathbb{R}^{J \times IK}$ and the mode-3 unfolding to $\mathbf{A}_{K \times IJ} \in \mathbb{R}^{K \times IJ}$ as shown in Fig. 4.4. For the mode-1, mode-2 and mode-3 unfolding, the frontal slices $\mathbf{A}_k \in \mathbb{R}^{I \times J}$, the transposed frontal slices $\mathbf{A}_k^{\top} \in \mathbb{R}^{J \times I}$, and the transposed lateral slices $\mathbf{A}_j^{\top} \in \mathbb{R}^{K \times I}$ are used, respectively. The unfolded modes and their corresponding variables are also refereed to as *slow* or *fast*, depending on how fast the index of the mode is incremented e.g. for the mode-1 unfolding, i and j are considered to be fast and k to be slow, as k is only incremented after every J-th column. In the context of this thesis, unfolding will usually been performed along the first mode, if not stated otherwise. The definition by Kolda and Bader [155] does not utilise the horizontal slices $\mathbf{A}_i$ of $\underline{\mathbf{A}}$, but an

alternative mode-3 unfolding can be defined by permutating the indices of $\mathbf{A}_{K\times IJ}$, leading to $\mathbf{A}_{K\times JI} \in \mathbb{R}^{K\times JI}$, consisting of the transposed horizontal slices $\mathbf{A}_i^\top \in \mathbb{R}^{J\times K}$.

The presented notation for three-way arrays can be extended straightforwardly to multi-way arrays in general and I refer to Cichocki et al. [45] for a further discussion on the notation of n-way arrays with $n > 3$.

Using this notation, a video sequence consisting of T frames with a spatial resolution of $U \times V$ pixel, can be described as a three-way array $\underline{\mathbf{S}} \in \mathbb{R}^{U\times V\times T}$, where we assume that the pixels are real-valued. Each frame can then be considered as the frontal slice $\mathbf{S}_t$ of $\underline{\mathbf{S}}$ with $\mathbf{S}_t \in \mathbb{R}^{U\times V}$ and s_{uv} representing the individual pixels of S_t. Assuming we have N different videos sequences, the set of all sequences is denoted as $\mathcal{S}$ with $\mathcal{S} = \{\underline{\mathbf{S}}_1, \cdots, \underline{\mathbf{S}}_N\}$ and $|\mathcal{S}| = N$, the set of calibration sequences as $\mathcal{S}_C$ and the set of validation sequences as $\mathcal{S}_V$. Note, that we assumed in the above definition of $\underline{\mathbf{S}}$ that all colour information for a pixel s_{uv} can be expressed by one real value. Even tough this is generally not done, and each colour component is usually considered separately that would require us to define $\underline{\mathbf{S}}$ as a four-way array $\underline{\mathbf{S}} \in \mathbb{R}^{U\times V\times T\times C}$ with an additional colour mode C, I use for simplicity the definition of video as a three-way array in this thesis.

We are, however, not interested in the video sequence itself, but rather in objective features describing its properties. We therefore extract for each frame $\mathbf{S}_t$ different features, resulting in a feature row vector $\mathbf{x}_t \in \mathbb{R}^{1\times M}$, where M represents the total number of extracted features and x_m the m-th feature in $\mathbf{x}_t$. Assuming the calibration set $\mathcal{S}_C$ consists of all N different video sequences S and therefore $\mathcal{S}_C = \mathcal{S}$, we can concatenate the feature vectors $\mathbf{x}_t$ of the t-th frame of each sequence into a feature matrix $\mathbf{X}_t \in \mathbb{R}^{N\times M}$, where the n-th row in the feature slice $\mathbf{X}_t$ corresponds to the feature row vector $\mathbf{x}_{nt}$ of the t-th frame of the n-th video sequence in the calibration set. Each feature matrix $\mathbf{X}_t$ can then be considered as the t-th frontal feature slice of the three-way feature array $\underline{\mathbf{X}} \in \mathbb{R}^{N\times M\times T}$, that represents the M features of the N video sequences S in the calibration set $\mathcal{S}_C$ for all T frames. Also each horizontal feature slice $\mathbf{X}_n \in \mathbb{R}^{M\times T}$ of $\underline{\mathbf{X}}$ describes the features for all frames of a video sequence S, thus providing a complete representation of S in the feature domain. If necessary, the features may be pooled along the temporal dimension t, resulting in a matrix $\mathbf{X} \in \mathbb{R}^{N\times M}$, where each row $\mathbf{x}$ represents the feature vectors $\mathbf{x}_t$ pooled over all frames T. In addition to the objective features, the column vector $\mathbf{y} \in \mathbb{R}^{N\times 1}$ represents the visual quality of each of the N video sequences that was gained in subjective testing and the corresponding quality predictions for the N video sequences are denoted by the column vector $\widehat{\mathbf{y}} \in \mathbb{R}^{N\times 1}$. Fig. 4.5 on the next page illustrates the notation for the video sequences, the corresponding features and the visual quality. Note, that in practical implementations, both the pixels s_{uv} and the extracted features x_m of a video sequence's frame can usually be represented by floating point variables, justifying the assumption that both the pixels and the features are real-valued.

As an alternative to the notation discussed so far, a MATLAB-like notation can be used: the element a_i of a vector $\mathbf{a}$ is denoted as $\mathbf{a}(i)$, the element a_{ij} of a two-way array $\mathbf{A}$ as $\mathbf{A}(i,j)$, and the i-th row and j-th column of $\mathbf{A}$ as $\mathbf{A}(i,:)$ and $\mathbf{A}(:,j)$, respectively. For a three-way array $\underline{\mathbf{A}}$, an element a_{ijk} is denoted as $\underline{\mathbf{A}}(i,j,k)$. The (j,k)-th column, the (i,k)-th

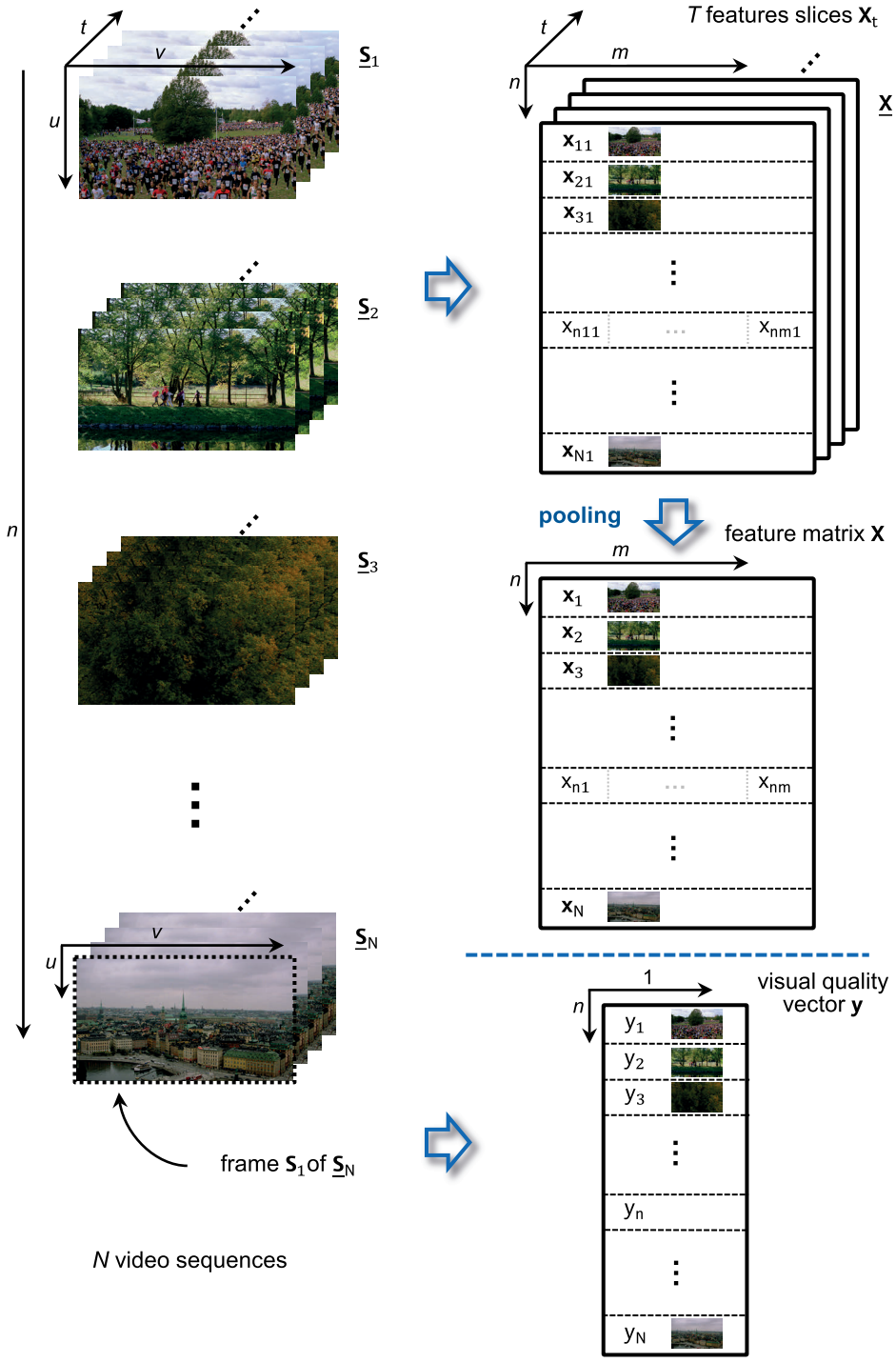

Figure 4.5.: Using the presented notation to describe video sequences **S**, their corresponding features **X** and the visual quality **y**

row and the (i, j)-th tube of $\underline{\mathbf{A}}$ are denoted as $\underline{\mathbf{A}}(:, j, k)$, $\underline{\mathbf{A}}(i, :, k)$ and $\underline{\mathbf{A}}(i, j, :)$, respectively. Similarly, the i-th horizontal, the j-th lateral and the k-th frontal slice are denoted as $\underline{\mathbf{A}}(i, :, :)$, $\underline{\mathbf{A}}(:, j, :)$ and $\underline{\mathbf{A}}(:, :, k)$, respectively. The indices of this MATLAB-like notation can also be moved from the parentheses into the subscript as proposed by Cichocki et al. in [45], e.g. the k-th frontal slice is then denoted as $\underline{\mathbf{A}}_{::k}$. Even though the MATLAB-like notation has the advantage of avoiding any ambiguities per definition, it can be cumbersome, thus reducing the clarity in equations and is therefore avoided as much as possible in this thesis. Kolda and Bader [155] also propose an alternative notation for the mode-1, mode-2 and mode-3 unfolding as $\mathbf{A}_{(1)}$,$\mathbf{A}_{(2)}$ and $\mathbf{A}_{(3)}$, respectively. Although more compact, it is, however, not as illustrative as the notation used in this thesis.

Although the aim in this thesis is to name each variable as unambiguously as possible, the use of well established notations from literature for certain methods or algorithms may cause ambiguities in some parts of this thesis. Where necessary, I clarify such ambiguities in the corresponding text.

4.2.2. Preprocessing of the data

Before we can perform the data analysis on the extracted features and the subjective visual quality, the data needs to be preprocessed. Often the data contains a constant offset and the individual features may exhibit different variance due the used scales or noise. Both offset and different variance, however, may influence the model building negatively, leading to models with reduced prediction performance. These two issues can be handled by *centring* and *scaling* of the data in a preprocessing step. Considering multi-way arrays in general, the centring and scaling can be performed on different modes of the array.

Centring describes the removal of any existing constant offset across a selected mode by subtracting the mean over all fibres of the selected mode from each individual fibre of the selected mode in the data. Hence, the structure of the data is changed by the offset removal. Although there are other definitions of centring in literature, centring refers always to mean-centring in this thesis. Based on the statement by Harshman and Lundy [83] that often subjective reasons are provided for performing centring, Smilde et al. [298] propose that centring can objectively be justified if centring makes a difference in the model by reducing the rank of the (component) model, an increased fit to the data or by avoiding numerical problems. Hence, centring should only be performed on the condition that offsets are present in the data and that the removal of the offsets leads to an improvement.

Scaling is the transformation of the fibres within a selected mode by multiplying all fibres within the selected mode by a certain weight. It does not change the structure of the data itself, but rather changes the importance of certain parts of the data in the model fitting. As data analysis methods often assume that a feature with a larger variance is more important, problems can arise if the large variance of a certain features is not caused by the features' influence itself, but rather by noise or by the use of a different scale compared to other features used in the model building. In practice, scaling usually aims to reduce the influence of different scales and the features are therefore standardised to unit vari-

ance. This is also the definition of scaling used in this thesis. Alternatively, a scaling to similar noise levels can be performed, if further information about the features' noise is available [197]. Similar to centring, scaling to unit variance should only be done if it leads to an improved model. If the features are nearly constant and therefore exhibit only a very small variance, the scaling will increase the magnitude of the features significantly. Especially if this small variation is only caused by noise, the scaling will consequently lead to a disproportional influence of the noise in the model [195, 362].

Scaling and centring can be performed on any mode of a multi-way array. Often the centring is performed across the first mode and scaling within the second mode. This combination of mode-1 centring and mode-2 scaling is also referred to as *autoscaling*. It aims at removing the offset from the first mode that represents the samples and at scaling the features within the second mode to unit variance. Centring and scaling, however, are not independent from each other [15]: centring across one mode has an influence on the scaling within all modes unless the scaling is to unit variance, whereas scaling within one mode only influences the centring across the same mode. Therefore centring is usually performed before scaling. In addition, centring across and scaling within the same mode will generally not retain the desired properties of the individual operations [32]. The notation of scaling *within* and centring *across* is motivated by Smilde et al. in [298] by the fact that for scaling all elements *within* the selected mode are multiplied by the same scalar and for centring the same offset is subtracted from all elements *across* the selected mode.

Considering the three-way feature array $\underline{\mathbf{X}}$ with its three modes, the first mode represents the different video sequences corresponding to the different samples, the second mode the extracted features and the third mode the different frames of the video sequences. Assuming that the extracted features include an offset for all video sequences and that the magnitude of the different features vary, autoscaling is a suitable preprocessing. Although we could also perform a mode-3 scaling along the temporal mode, the variation of the features along the temporal mode will be within the same variance after the mode-2 scaling, as we extract the same features for all frames of the video sequences. If the three-way feature array $\underline{\mathbf{X}}$ is pooled along its temporal mode into a two-way feature array $\mathbf{X}$, autoscaling is also applied to $\mathbf{X}$ after the pooling operation.

Preprocessing of two-way arrays For a two-way feature array $\mathbf{X}$, the mode-1 centring can be described for each element x_{nm} as

$$\overline{x}_{nm} = x_{nm} - \frac{1}{N}\sum_{n=1}^{N} x_{nm}, \tag{4.2.1}$$

where $\overline{x}_{nm}$ describes an element of the centred feature matrix $\overline{\mathbf{X}}$. With $\overline{\mathbf{x}} \in \mathbb{R}^{M\times 1}$ as the vector containing the average of the m-th column in the in its m-th element

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{X}, \tag{4.2.2}$$

and $\mathbf{1} \in \mathbb{R}^{N\times 1}$ a vector of ones, the centring of $\mathbf{X}$ and the resulting centred featured matrix $\overline{\mathbf{X}}$ can be expressed as

$$\overline{\mathbf{X}} = \mathbf{X} - \mathbf{1}\overline{\mathbf{x}}^{\top}. \tag{4.2.3}$$

The mode-2 scaling of the two-way feature array $\mathbf{X}$ to unit variance for each element x_{nm} of $\mathbf{X}$ of is given as

$$z_{nm} = x_{nm}\frac{1}{\sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(x_{nm}-\overline{x}_{nm})^2}}, \tag{4.2.4}$$

where z_{nm} describes an element of the scaled feature matrix $\mathbf{Z}$. With $\mathbf{w} \in \mathbb{R}^{M\times 1}$ as the vector containing the inverse sample standard deviation of the m-th column in its m-th element w_m

$$w_m = \frac{1}{\sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(x_{nm}-\overline{x}_{nm})^2}} \tag{4.2.5}$$

and the diagonal matrix $\mathbf{W} \in \mathbb{R}^{M\times M}$

$$\mathbf{W} = \mathbf{I}_M\mathbf{w}, \tag{4.2.6}$$

where the m-th diagonal element corresponds to the inverse sample standard deviation of the m-th column and we used the identity matrix $\mathbf{I}_M \in \mathbb{R}^{M\times M}$, the mode-2 scaling can be expressed as

$$\mathbf{Z} = \mathbf{X}\mathbf{W}. \tag{4.2.7}$$

Preprocessing of three-way arrays Extending centring and scaling from two-way arrays to three-way arrays, the mode-1 centring for the three-way feature array $\underline{\mathbf{X}}$ can be described for each element x_{nmt} as

$$\overline{x}_{nmt} = x_{nmt} - \frac{1}{N}\sum_{n=1}^{N}x_{nmt}, \tag{4.2.8}$$

where $\overline{x}_{nmt}$ describes an element of the centred feature matrix $\underline{\overline{\mathbf{X}}}$. With $\mathbf{M} \in \mathbb{R}^{M\times T}$ as the matrix containing the average of the (m,t)-th column of $\underline{\mathbf{X}}$ in the in its (m,t)-th element

$$\mathbf{M} = \frac{1}{N}\sum_{n=1}^{N}\underline{\mathbf{X}}, \tag{4.2.9}$$

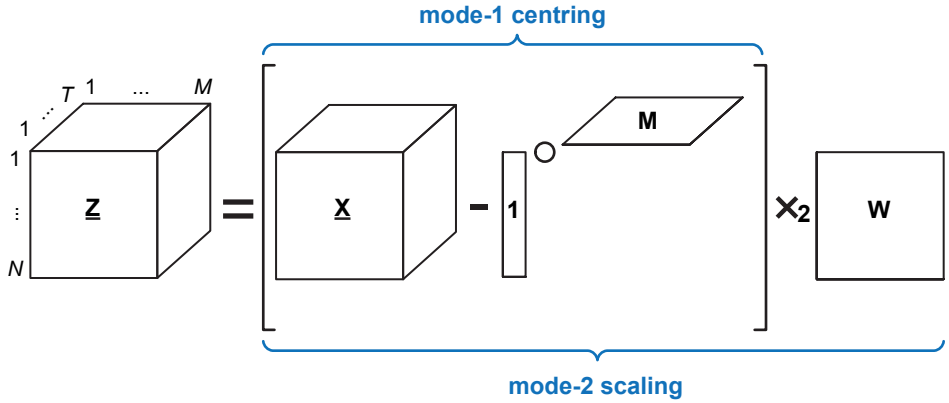

Figure 4.6.: Preprocessing of the three-way feature array $\underline{\mathbf{X}}$: mode-1 centring and mode-2 scaling

and $\mathbf{1} \in \mathbb{R}^{N\times 1}$ a vector of ones, the centring of $\underline{\mathbf{X}}$ and the resulting centred featured matrix $\overline{\underline{\mathbf{X}}}$ can be written as

$$\overline{\underline{\mathbf{X}}} = \underline{\mathbf{X}} - \mathbf{1} \circ \mathbf{M}, \tag{4.2.10}$$

Note, that instead of subtracting a vector of column averages as in the two-way case, we subtract a matrix of column averages for three-way arrays. Alternatively, the mode-1 centring of the three-way $\underline{\mathbf{X}}$ can be done by first performing the mode-1 unfolding of $\underline{\mathbf{X}}$ into $\mathbf{X}_{N\times MT}$ and then applying the mode-1 centring for two-way arrays as described before [32]. We can therefore also write (4.2.10) as

$$\overline{\mathbf{X}}_{N\times MT} = \mathbf{X}_{N\times MT} - \mathbf{1}\overline{\mathbf{x}}^{\top}_{N\times MT}, \tag{4.2.11}$$

where $\overline{\mathbf{x}}_{N\times MT} \in \mathbb{R}^{MT\times 1}$ is a vector containing the average of the mt-th column in its mt-th element, that can be determined in the same way as in (4.2.2).

Similar as for two-way arrays, the mode-2 scaling of the three-way feature array $\underline{\mathbf{X}}$ to unit variance for each element x_{nmt} of $\underline{\mathbf{X}}$ is given as

$$z_{nmt} = x_{nmt} \frac{1}{\sqrt{\frac{1}{(N-1)(T-1)} \sum_{n=1}^{N} \sum_{t=1}^{T} (x_{nmt} - \overline{x}_{nmt})^2}}, \tag{4.2.12}$$

where z_{nmt} describes an element of the scaled feature array $\underline{\mathbf{Z}}$. With $\mathbf{w} \in \mathbb{R}^{M\times 1}$ as the vector containing the inverse sample standard deviation of the m-th lateral slice $\mathbf{X}_m$ in its m-th element w_m

$$w_m = \frac{1}{\sqrt{\frac{1}{(N-1)(T-1)} \sum_{n=1}^{N} \sum_{t=1}^{T} (x_{nmt} - \overline{x}_{nmt})^2}} \tag{4.2.13}$$

we can define the diagonal matrix $\mathbf{W} \in \mathbb{R}^{M\times M}$

$$\mathbf{W} = \mathbf{I}_M \mathbf{w}, \tag{4.2.14}$$

where the m-th diagonal element corresponds to the inverse sample standard deviation of the m-th lateral slice. After unfolding $\underline{\mathbf{X}}$ along the second mode into $\mathbf{X}_{M\times NT}$, the mode-2 scaling of $\underline{\mathbf{X}}$ can then be expressed as

$$\mathbf{Z}_{M\times NT} = \mathbf{W}\mathbf{X}_{M\times NT}, \tag{4.2.15}$$

where $\mathbf{Z}_{M\times NT}$ can easily be rearranged back into the scaled three-way feature array $\underline{\mathbf{Z}}$. By using the *mode-2 product* $\times_2$, we can rewrite (4.2.15) without unfolding as [155]

$$\underline{\mathbf{Z}} = \underline{\mathbf{X}} \times_2 \mathbf{W}. \tag{4.2.16}$$

Both the mode-1 centring and the mode-2 scaling of the three-way feature array $\underline{\mathbf{X}}$ are illustrated in Fig. 4.6 on the facing page.

Similarly to the features, also the visual quality vector $\mathbf{y}$ needs to be preprocessed. As $\mathbf{y}$ is a one-way array, representing only one variable and is therefore univariate, scaling is not necessary and only mode-1 centring in order to remove any offset is performed. For each sequences' visual quality y_n the centred visual quality $\overline{y}_n$ can be expressed as

$$\overline{y}_n = y_n - \frac{1}{N}\sum_{n=1}^{N} y_n, \tag{4.2.17}$$

and with $\overline{y}$ as the average visual quality of all sequences as

$$\overline{y} = \frac{1}{N}\sum_{n=1}^{N} y_n, \tag{4.2.18}$$

we can write the mode-1 centring of $\mathbf{y}$ as

$$\overline{\mathbf{y}} = \mathbf{y} - \mathbf{1}\overline{y}, \tag{4.2.19}$$

where $\mathbf{1} \in \mathbb{R}^{N\times 1}$ is again a vector of ones.

If not stated otherwise, the feature arrays $\mathbf{X}$ and $\underline{\mathbf{X}}$ are assumed to be mode-1 mean centred and mode-2 scaled to standard deviation, and the subjective quality vector $\mathbf{y}$ is assumed to be mode-1 mean centred in this thesis. For a more detailed discussion on centring and scaling in general, but especially in the context of multi-way analysis, I refer to Bro and Smilde [32] and in Smilde et al. [298].

5. Two-way Data Analysis

In the design of video quality metrics, so far mostly (linear) two-way data analysis methods have been applied to two-way feature arrays, generated from the three-way feature arrays by temporal pooling. I provide therefore in this chapter an overview of these methods by discussing different two-way data analysis methods and their corresponding regression models: the multiple linear regression, principal component regression and partial least squares regression. Additionally, the temporal pooling that is necessary to prepare the features for the application of these methods is discussed in detail.

5.1. Temporal pooling

Before two-way data analysis methods can be applied to the extracted features and the subjective visual quality, we need to generate a two-way feature array $\mathbf{X}$ from the originally extracted three-way feature array $\underline{\mathbf{X}}$, where each frontal slice $\mathbf{X}_t$ represents the features of the t-th frame $\mathbf{S}_t$ of the video sequence $\underline{\mathbf{S}}_n$ in $\mathcal{S}_C$. The aim is to map the temporal variation within each feature of a video sequence into a single value x_{nm} by applying an appropriate pooling function $f_P : \mathbb{R}^{N\times M\times T} \rightarrow \mathbb{R}^{N\times M}$, $f_p(\underline{\mathbf{X}}) = \mathbf{X}$, that captures the variation over time sufficiently well. This process is called *temporal pooling* as it is performed along the temporal dimension of $\underline{\mathbf{X}}$ as illustrated in Fig. 5.1 on the next page. Note that the preprocessing as described in the previous chapter is performed after the temporal pooling i.e. on the two-way array $\mathbf{X}$ instead of the three-way array $\underline{\mathbf{X}}$.

In the context of visual quality metrics, the most commonly used pooling function is the averaging by calculating the arithmetic mean of the features over all T frames

$$\mathbf{X} = f_P(\underline{\mathbf{X}}) = \underset{T}{\text{mean}}(\underline{\mathbf{X}}) = \frac{1}{T}\sum_{t=1}^{T}\underline{\mathbf{X}}, \tag{5.1.1}$$

where each element x_{nm} of $\mathbf{X}$ is given by

$$x_{nm} = \frac{1}{T}\sum_{t=1}^{T} x_{nmt}. \tag{5.1.2}$$

Other pooling functions aim to capture different statistical properties of the temporal variation of the features and are applied similar to the arithmetic mean in (5.1.1): the *median*, *standard deviation*, *minimum*, *maximum*, *10%-percentile* and *90%-percentile* [367]. Sometimes a combination of these pooling function is used for each feature i.e. for each feature

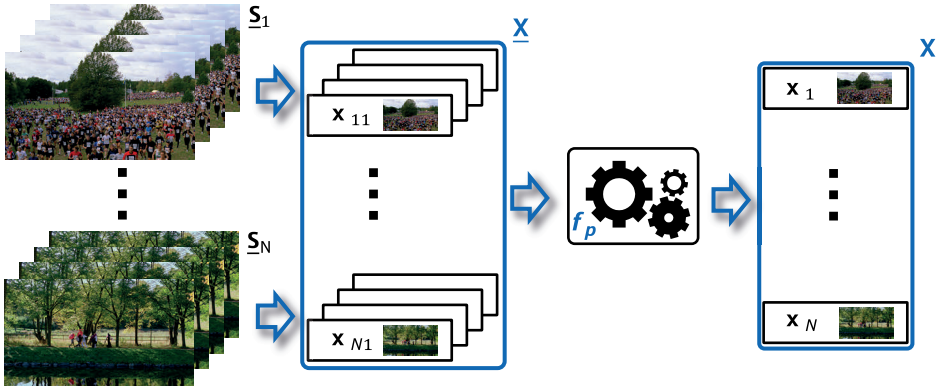

Figure 5.1.: Temporal pooling of the three-way feature array $\underline{\mathbf{X}}$ consisting of T feature vectors $\mathbf{x}_t$ for each of the N video sequence $\underline{\mathbf{S}}_n$ with a pooling function f_P, resulting in a two-way feature array $\mathbf{X}$, where each row $\mathbf{x}_n$ represents the pooled features of one sequence

multiple statistical properties of its temporal variation are included in the two-way feature array $\mathbf{X}$ as additional columns $\mathbf{x}_m$, leading to a larger two-feature array $\mathbf{X}_p \in \mathbb{R}^{N \times P}$ with $P > M$. In my previous contributions [143, 145], I used these additional simple statistical functions to increase the prediction performance compared to averaging over all frames as in (5.1.1) of both pixel- and bitstream-based no reference video quality metrics built with two-way data analysis methods. The statistical functions capturing the temporal variation the best, however, are depending very strongly on the individual features as shown in [143, 145]. Hence, it is not possible to select a general set of statistical pooling functions, but rather they need to be selected depending on the actual features. As this thesis aims to cover the design of video quality metrics in a more general approach, I therefore consider only temporal pooling by averaging over all frames T in this thesis, even though simple averaging may not be the optimal temporal pooling strategy.

Another popular pooling function is the application of the normalised L_p norm to $\underline{\mathbf{X}}$ along the temporal dimension over all frames, where each element x_{nm} of the pooled feature array $\mathbf{X}$ is expressed as

$$x_{nm} = \frac{1}{T}\left(\sum_{t=1}^{T} {x_{nmt}}^p\right)^{1/p}, \tag{5.1.3}$$

with p usually between $p = 2$ and $p = 5$. Especially for temporal pooling, Winkler [354] suggests to use $p = 4$. The normalised L_p norm is usually referred to as *Minkowski summation* in the research community on vision. The use of the L_p norm was first proposed by Quick [262] and was motivated by its adequate description of the probability summation process of different detection channels in the HVS. Since then, it has been frequently used for both spatial and temporal pooling in contributions to video quality metrics, e.g. by de Ridder [269], Watson et al. [343], Winkler [354]. You et al. [372], however, recently suggested that

while Minkowski summation is suitable for spatial pooling, it may not be an appropriate pooling scheme for temporal pooling.

5.2. Multiple linear regression (MLR)

The classic approach to two-way array data analysis is the *univariate multiple linear regression (MLR)*. *Multiple* denotes that more than one independent variable is used in the regression i.e. $m > 1$ and $\mathbf{X}$ is therefore a two-way array, whereas *univariate* denotes that the dependent variable is a vector and not a two-way array.

Considering the desired linear model between the two-way feature array $\mathbf{X} \in \mathbb{R}^{N \times M}$ and the subjective visual quality vector $\mathbf{y} \in \mathbb{R}^{N \times 1}$, representing the sequences in the calibration set $\mathcal{S}_C$

$$\mathbf{y} = \mathbf{X}\mathbf{b}, \tag{5.2.1}$$

where $\mathbf{b} \in \mathbb{R}^{M \times 1}$ represents the vector of weights for the individual features, the easiest way to solve the equation for $\mathbf{b}$ would be to multiply $\mathbf{y}$ with the inverse $\mathbf{X}^{-1}$ and thus

$$\mathbf{b} = \mathbf{X}^{-1}\mathbf{y}. \tag{5.2.2}$$

Obviously, $\mathbf{X}$ must be square and non-singular, for $\mathbf{X}^{-1}$ to exist. We can, however, neither assume that $\mathbf{X}$ is square, as usually $n \neq m$, nor that $\mathbf{X}$ is non-singular, as as we can not assume that columns of $\mathbf{X}$ representing the different features are linearly independent and therefore $\mathbf{X}$ would be singular.

The weight vector $\mathbf{b}$ is therefore replaced by its least square estimator $\widehat{\mathbf{b}}$, that solves the least squares problem of minimising the least squares error between $\mathbf{y}$ and the model of $\mathbf{y}$ given by $\mathbf{X}\mathbf{b}$

$$\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2, \tag{5.2.3}$$

as

$$\widehat{\mathbf{b}} = \mathbf{X}^{+}\mathbf{y}, \tag{5.2.4}$$

where $\mathbf{X}^{+}$ denotes the Moore-Penrose pseudoinverse of $\mathbf{X}$ [244, 264]. Hence $\widehat{\mathbf{b}}$ represents the solution for the optimisation problem in (5.2.3). In the general case, $\mathbf{X}^{+}$ can be determined with the *singular value decomposition (SVD)* $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^{\top}$ of $\mathbf{X}$ as

$$\mathbf{X}^{+} = \mathbf{V}\mathbf{S}^{+}\mathbf{U}^{\top}, \tag{5.2.5}$$

where s^{+}_{nn} is the n-th diagonal element of $\mathbf{S}^{+}$ defined as

$$s^{+}_{nn} = \begin{cases} \frac{1}{s_{nn}} & \text{if } s_{nn} \neq 0 \\ 0 & \text{if } s_{nn} = 0. \end{cases} \tag{5.2.6}$$

For the special case that the features in the columns of $\mathbf{X}$ are linear independent and therefore $\mathbf{X}$ has full rank, $\mathbf{X}^{+}$ is given by

$$\mathbf{X}^{+} = (\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}^{\top}, \tag{5.2.7}$$

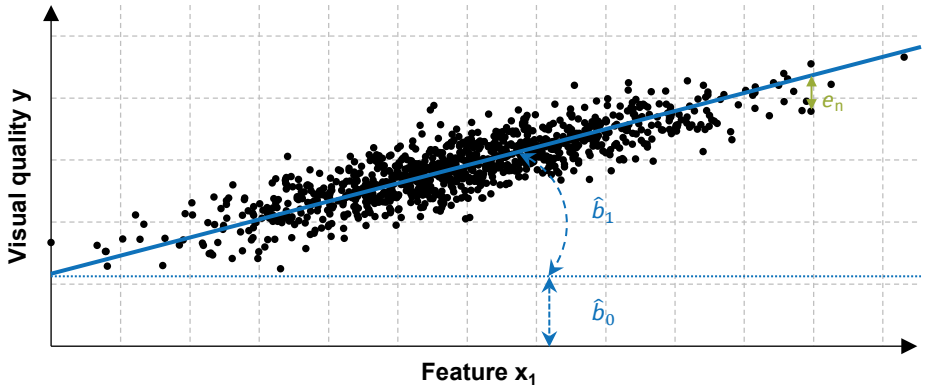

Figure 5.2.: Multiple linear regression: qualitative interpretation of $\mathbf{y}$ and $\widehat{\mathbf{b}}$ for the first feature $\mathbf{x}_1$: for the $n-$th video sequence the visual quality y_n is related to the corresponding feature $x_{1,n}$ by $y_n = x_{1,n}\widehat{b}_1 + \widehat{b}_0 + e_n$

and if $\mathbf{X}$ is square and non-singular $\mathbf{X}^+ = \mathbf{X}^{-1}$. For a further discussion of the properties of the Moore-Penrose pseudoinverse, I refer to Campbell and Meyer [36].

The predicted visual quality vector $\widehat{\mathbf{y}}$ for the N sequences in the calibration set $\mathcal{S}_C$ is then given as

$$\widehat{\mathbf{y}} = \mathbf{X}\widehat{\mathbf{b}}, \tag{5.2.8}$$

and the true visual quality vector $\mathbf{y}$ can be written as

$$\mathbf{y} = \mathbf{X}\widehat{\mathbf{b}} + \mathbf{e}, \tag{5.2.9}$$

with $\mathbf{e}$ as the error term or *residual* of the model caused by noise and modelling errors. The relationship between $\mathbf{y}$ and $\widehat{\mathbf{b}}$ is illustrated in Fig. 5.2. For an unknown video sequence $\underline{\mathbf{S}}_U \notin \mathcal{S}_C$ and the corresponding feature vector $\mathbf{x}_u \in \mathbb{R}^{1\times M}$, we are then able to predict its visual quality $\widehat{y}_u$ with

$$\widehat{y}_u = \mathbf{x}_u\widehat{\mathbf{b}} + \widehat{b}_0 \tag{5.2.10}$$

where it is assumed that $\mathbf{x}_u$ has been preprocessed with the same parameters as $\mathbf{X}$ and the resulting subjective quality prediction $\widehat{y}_u$ needs to be corrected by the offset $\widehat{b}_0 = \overline{y}$, as the visual quality vector $\mathbf{y}$ has been mean centred during calibration according to (4.2.17).

In order to predict the visual quality of unknown sequences without preprocessing of the feature vector $\mathbf{x}_u$, the weight vector $\widehat{\mathbf{b}}$ and the offset $\widehat{b}_0$ can be modified to take the centring in scaling directly into account. From

$$\widehat{y}_u = ((\mathbf{x}_u - \overline{\mathbf{x}})\mathbf{W})\widehat{\mathbf{b}} + \overline{y}, \tag{5.2.11}$$

where $\overline{\mathbf{x}}$ and $\mathbf{W}$ are the vector of the column averages of $\mathbf{X}$ according to (4.2.2) and the scaling weights for the individual features according to (4.2.6), respectively, we can deter-

mine a new weight vector $\widehat{\mathbf{b}}$ and offset $\widehat{b}_0$ as

$$\widehat{\mathbf{b}} = \mathbf{W}\mathbf{b} \tag{5.2.12a}$$

and

$$\widehat{b}_0 = \overline{y} - \overline{\mathbf{x}}\mathbf{W}\mathbf{b}. \tag{5.2.12b}$$

The visual quality for unknown video sequences without preprocessing of $\mathbf{x}_u$ is then given by

$$\widehat{y}_u = \mathbf{x}_u\widehat{\mathbf{b}} + \widehat{b}_0. \tag{5.2.13}$$

Although we will see in Chapter 9 that the prediction performance for sequences in the calibration set is very good, the prediction performance for unknown video sequences is rather bad and thus the model is not stable for sequences not included in the calibration set $\mathcal{S}_C$. This can be explained by the fact that some of the columns of $\mathbf{X}$ representing the different features may be approximately or exactly linearly dependent and $\mathbf{X}$ is then called *collinear*. It can be shown [197] that this leads to a large variance for some elements in $\widehat{\mathbf{b}}$, as in this case the least squares solution is strongly influenced by noise in the data [13]. Using the equivalence of the SVD of $\mathbf{X}$ to the eigendecomposition of $\mathbf{X}^\top\mathbf{X}$, this can also be expressed by how well the new unknown video sequence fits into the range of variability of the calibration samples along the different eigenvector axes, where the fit has a tendency to be poorer for small eigenvalues [223]. Consequently, the collinearity may have a negative effect on the stability of the MLR model, leading to high variance of the quality prediction $\widehat{\mathbf{y}}$ for unknown video sequences, that were not included in the calibration set $\mathcal{S}_C$.

Another problem is that 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 or combination of features may describe the variation of the visual quality better and thus may contain more information than others.

Lastly, the a-priori understanding of the relationships between features and visual quality may be inadequate, as the sequences in the calibration set $\mathcal{S}_C$ may not represent the general population sufficiently enough in order to derive the influence of some features in the model building with MLR. Martens and Næs discuss these three issues of *collinearity*, *lack of selectivity* and *lack of knowledge* for linear models in more detail in [197].

5.3. Component models

The instability problem of the model building with MLR described in the previous section can be addressed if not the features are used directly in the model building, but rather so-called *latent variables* or *components* that represent a hidden combination of the features are used. In other words, we aim to reduce or compress the dimensionality of our original feature space into a more compact representation, more fitting for our latent variables. The assumption is that these new, more compact representation of the features capture the influence of the features on the quality variation better and thus lead to models that are applicable more ubiquitously. As we are modelling the latent variables, Burnham et al. [35] refer to this general approach as *latent variable multivariate regression* (LVMR) modelling.

Martens and Næs formulate in [197] the general framework of this approach for the calibration as

$$\widehat{\mathbf{V}} = f_C(\mathbf{X}, \mathbf{Y}) \tag{5.3.1a}$$

$$\widehat{\mathbf{T}} = \mathbf{X}\widehat{\mathbf{V}} \tag{5.3.1b}$$

$$\mathbf{X} = \widehat{\mathbf{T}}\mathbf{P}^\top + \mathbf{E} \tag{5.3.1c}$$

$$\mathbf{Y} = \widehat{\mathbf{T}}\mathbf{Q}^\top + \mathbf{F}, \tag{5.3.1d}$$

where $\widehat{\mathbf{V}}$ is a weight matrix gained by optimising a chosen criterion with respect to the cost function f_C over $\mathbf{X}$ and $\mathbf{Y}$, and $\mathbf{E}$ and $\mathbf{F}$ represent the residual for $\mathbf{X}$ and $\mathbf{Y}$, respectively. As both $\mathbf{X}$ and $\mathbf{Y}$ are two-way arrays in this general framework, such a component model is also called a *two block* model. The notation of $\widehat{\mathbf{V}}$ and $\widehat{\mathbf{T}}$ is motivated in [197] by the fact that our calibration set is in most cases only a subset of the complete population and therefore only an estimate of the indeterminable real $\mathbf{V}$ and $\mathbf{T}$. As we approximate $\mathbf{X}$ by a product of two sets of estimated linear parameters in (5.3.1c), $\mathbf{P}$ and $\widehat{\mathbf{T}}$, this general method is called *bilinear modelling*. Note, that (5.3.1c) and (5.3.1d) contain a term describing the model and an error term of the part of the $\mathbf{X}$ and $\mathbf{Y}$, indicating that the model describes $\mathbf{X}$ and $\mathbf{Y}$ not perfectly. The *scores* $\widehat{\mathbf{T}}$ in (5.3.1b) are a representation of $\mathbf{X}$ with respect to the new basis $\widehat{\mathbf{V}}$ defined by the optimisation criterion, and the *loadings* $\mathbf{P}$ and $\mathbf{Q}$ represent the weights with respect to the scores $\widehat{\mathbf{T}}$ for $\mathbf{X}$ and $\mathbf{Y}$, respectively. The loadings can also be considered as a projection of the original variables in $\mathbf{X}$ and $\mathbf{Y}$ onto new subspaces in which $\mathbf{X}$ and $\mathbf{Y}$ can be represented by the scores $\widehat{\mathbf{T}}$. $\widehat{\mathbf{T}}$ contains in its columns $\widehat{\mathbf{t}}$ the components in order of their influence on the optimisation criterion i.e. the first column $\widehat{\mathbf{t}}_1$ represents the component with the strongest influence, the second column $\widehat{\mathbf{t}}_2$ the component with the second strongest influence etc. The least squares estimate of $\mathbf{P}$ and $\mathbf{Q}$ as $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$ is then given by

$$\widehat{\mathbf{P}}^\top = (\widehat{\mathbf{T}}^\top\widehat{\mathbf{T}})^{-1}\widehat{\mathbf{T}}^\top\mathbf{X}^\top \tag{5.3.2a}$$

$$\widehat{\mathbf{Q}}^\top = (\widehat{\mathbf{T}}^\top\widehat{\mathbf{T}})^{-1}\widehat{\mathbf{T}}^\top\mathbf{Y}^\top, \tag{5.3.2b}$$

where it is assumed that the inverse of $\widehat{\mathbf{T}}^\top\widehat{\mathbf{T}}$ exists. Note that we first obtain the scores $\widehat{\mathbf{T}}$ according to the optimisation criterion defined with f_c and then find the corresponding subspaces defined by $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{Q}}$.

Using the feature vector of $\mathbf{x}_u$ of an unknown sample U, we are then able to predict the corresponding $\widehat{\mathbf{y}}_u$. We first obtain the scores $\widehat{\mathbf{t}}_u$ of $\mathbf{x}_u$ as

$$\widehat{\mathbf{t}}_u = \mathbf{x}_u\widehat{\mathbf{V}} \tag{5.3.3}$$

and then predict $\widehat{\mathbf{y}}_u$ with

$$\widehat{\mathbf{y}}_u = \widehat{\mathbf{t}}_{u,R}\widehat{\mathbf{Q}}^\top, \tag{5.3.4}$$

where $\widehat{\mathbf{t}}_{u,R}$ contains only the first R entries of $\widehat{\mathbf{t}}_u$, representing the first R components as described before. We therefore chose only the R most important components for our prediction model, assuming that the less important components mainly describe noise. For a direct prediction of $\widehat{\mathbf{y}}_u$ without calculating the scores, we can write (5.3.4) as

$$\widehat{\mathbf{y}}_u = \mathbf{x}_u\widehat{\mathbf{B}}, \tag{5.3.5}$$

where

$$\widehat{\mathbf{B}} = \widehat{\mathbf{V}}\widehat{\mathbf{Q}}^\top, \tag{5.3.6}$$

which follows straightforwardly from (5.3.3) and (5.3.4). For the clarity, the circumflex in the notation for the scores and loadings will be omitted from now on.

5.4. Principle component regression (PCR)

One method utilising a component model is the principal component regression (PCR). It combines the principal component analysis (PCA) for extracting the the principal components with a subsequent regression of in component model as described in the previous section. One of the first to propose the idea behind the PCA was Pearson [242], but the method itself was first suggested by Fisher and Mackenzie [64] and the name principal component analysis was introduced by Hotelling [90].

The aim of PCA can be considered from two different viewpoints [298]: the *variance of the principal components* and *variance explained by the principal components*. Before introducing the PCR, I briefly discuss these two interpretations and refer to Smilde et al. [298] for more information.

Variance of the principal components This approach aims to express the feature matrix $\mathbf{X} \in \mathbb{R}^{N\times M}$ as a linear combination of scores $\mathbf{t}_r \in \mathbb{R}^{N\times 1}$ and weights $\mathbf{w}_r \in \mathbb{R}^{M\times 1}$ with $i = 1 \ldots I$ representing the index of the principal components, this can be expressed as

$$\mathbf{t}_i = \mathbf{X}\mathbf{w}_i, \tag{5.4.1}$$

where the score $\mathbf{t}_1$ has the highest variance and $\mathbf{t}_l$ the lowest variance. Under the condition that $\mathbf{w}_i$ is restricted to length one in order to avoid that the variance of $\mathbf{t}_i$ becomes arbitrarily large, the problem of finding the first score $\mathbf{t}_1$ with the highest variance is given as

$$\max_{\|\mathbf{w}_1\|=1} \text{var}(\mathbf{t}_1) \tag{5.4.2a}$$

which is equivalent due to the mean centred $\mathbf{X}$ to [298]

$$\max_{\|\mathbf{w}_1\|=1} (\mathbf{t}_1^\top \mathbf{t}_1), \tag{5.4.2b}$$

and with (5.4.1) we can rewrite (5.4.2b)

$$\max_{\|\mathbf{w}_1\|=1} (\mathbf{w}_1^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}_1) \tag{5.4.3}$$

and arrive at a standard optimisation problem with the solution [14]

$$\mathbf{t}_1 = \mathbf{X}\mathbf{v}_1 = \mathbf{X}\mathbf{u}_1 s_1, \tag{5.4.4}$$

where $\mathbf{v}_1$ is the first eigenvector of $\mathbf{X}^\top \mathbf{X}$ or equivalently the first right singular vector of $\mathbf{X}$, s_1 is the largest and first singular value of $\mathbf{X}$ and $\mathbf{u}_1$ is the first left singular vector of $\mathbf{X}$. In the next step, the second score $\mathbf{t}_2$ is gained equivalent to the first score $\mathbf{t}_1$ under the condition that $\mathbf{w}_1^\top \mathbf{w}_2 = 0$, leading to orthogonal $\mathbf{t}_1$ and $\mathbf{t}_2$. This is repeated for all other components.

Variance explained by the principal components The second approach goes back to the first ideas by Pearson [242] and tries to find a subspace that describes the original data best in a least square sense. Using scores $\mathbf{t}_i \in \mathbb{R}^{N\times 1}$ according to (5.4.4) as used before, $\mathbf{X} \in \mathbb{R}^{N\times M}$ can be regressed on the loadings $\mathbf{p}_i \in \mathbb{R}^{M\times 1}$, that describe how well each column $\mathbf{x}_m$ of $\mathbf{X}$ can be described by $\mathbf{t}_i$. For the i-th component, this is equivalent to

$$\min_{\mathbf{p}_i} \|\mathbf{X} - \mathbf{t}_i \mathbf{p}_i^\top\|^2, \tag{5.4.5}$$

with the solution

$$\mathbf{p}_i = \mathbf{X}^\top \mathbf{t}_i (\mathbf{t}_i^\top \mathbf{t}_i)^{-1}, \tag{5.4.6}$$

where we use that $\mathbf{t}_i$ is orthogonal and therefore the inverse of $\mathbf{t}_i^\top \mathbf{t}_i$ exists. Also it can be shown that $\mathbf{w}_i = \mathbf{p}_i = \mathbf{v}_i$ [14] and therefore (5.4.4) can be rewritten as

$$\mathbf{t}_i = \mathbf{X}\mathbf{p}_i. \tag{5.4.7}$$

The loadings vector $\mathbf{p}_i$ gives a direction of a subspace in the M-dimensional space and thus maps the original variables into a new coordinate system whereas the scores $\mathbf{t}_i$ represent the orthogonal projections on the coordinate axes in this subspace. This is illustrated for

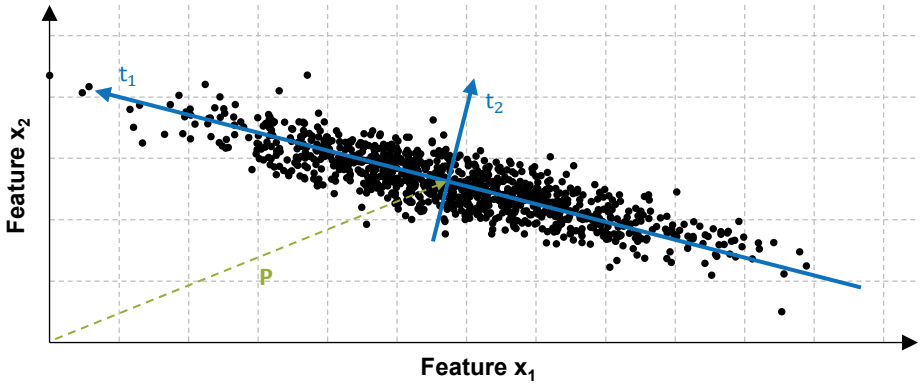

Figure 5.3.: Representing the features x_1 and x_2 of the n sequences with the scores t_1 and t_2 for the first two principal components and interpretation of the loadings P as a mapping of the original feature space into the a new subspace

two components in Fig. 5.3. Depending on the number of the R used principal components, this subspace is a line, a plane or a hyperplane for $R = 1$, $R = 2$ and $R \geq 3$, respectively.

Considering more than one component at a time results in $\mathbf{T}$ and $\mathbf{P}$ containing the scores and loadings for the R components as columns. We can then rewrite (5.4.5) as

$$\min_{\mathbf{T},\mathbf{P}} \|\mathbf{X} - \mathbf{T}\mathbf{P}^\top\|^2, \tag{5.4.8}$$

and with (5.4.1) we arrive at

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{P}^\top\|^2, \tag{5.4.9}$$

which is equivalent to the optimisation problem in (5.4.3) as once $\mathbf{W}$ is known, $\mathbf{P}$ can always be determined [298]. Note, that for the PCA the weight matrix $\widehat{\mathbf{V}}$ discussed in Section 5.3 corresponds to the loadings matrix $\mathbf{P}$ i.e. $\widehat{\mathbf{V}} = \mathbf{P}$.

Determining the principal components After this short discussion about the interpretation of the PCA, I present methods to determine the principal components and to build prediction models with the principal components in the rest of this section.

Applying the SVD on the two-way feature array $\mathbf{X} \in \mathbb{R}^{N \times M}$ and assuming that $n > m$ i.e. more sequences than features, we can decompose $\mathbf{X}$ as

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top, \tag{5.4.10}$$

where $\mathbf{U} \in \mathbb{R}^{N \times N}$ contains the I left singular vectors in its I orthonormal columns, $\mathbf{V} \in \mathbb{R}^{M \times M}$ contains the I right singular vectors in its orthonormal columns and $\mathbf{S} \in \mathbb{R}^{N \times M}$ is a diagonal matrix with the (non-negative) singular values of $\mathbf{X}$ on its diagonal in descending order form the largest singular value for $i = 1$ to the smallest singular value for $i = I$. For $m > n$ the SVD of $\mathbf{X}$ can be found by applying (5.4.10) on the transpose $\mathbf{X}^\top$.

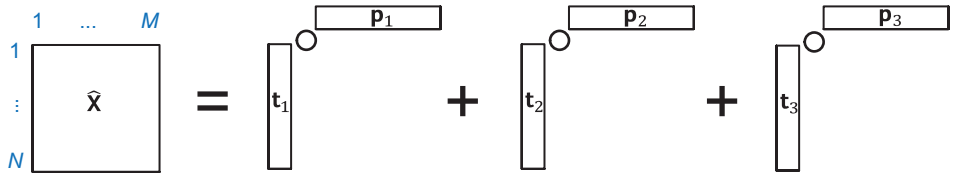

Figure 5.4.: Two-way component model for the rank-R approximation $\widehat{\mathbf{X}}$ of $\mathbf{X}$ for $R = 3$ with scores $\mathbf{t}_r$ and loadings $\mathbf{p}_r$

By not using all I singular values, but rather truncating after R components, with $R \leq I$, we can approximate $\mathbf{X}$ by $\widehat{\mathbf{X}}$ as

$$\widehat{\mathbf{X}} = \mathbf{U}_R \mathbf{S}_R \mathbf{V}_R^\top, \tag{5.4.11}$$

where $\mathbf{U}_R \in \mathbb{R}^{N \times R}$ contains the first R left singular vectors in its columns, $\mathbf{V}_R \in \mathbb{R}^{M \times R}$ contains the first R right singular vectors in its columns and $\mathbf{S}_R \in \mathbb{R}^{R \times R}$ is a diagonal matrix with the first R singular values of $\mathbf{X}$ on its diagonal in descending order form the largest singular value for $r = 1$ to the smallest singular value for $r = R$. $\widehat{\mathbf{X}}$ is thus also the best rank-R decomposition of $\mathbf{X}$ and we can write $\mathbf{X}$ as

$$\mathbf{X} = \widehat{\mathbf{X}} + \mathbf{E}_R, \tag{5.4.12}$$

where $\mathbf{E}_R$ is the residual of the rank-R approximation of $\mathbf{X}$ by $\widehat{\mathbf{X}}$ i.e. the part of $\mathbf{X}$ that is only modelled insufficiently by the R principal components and $\widehat{\mathbf{X}}$. With the scores

$$\mathbf{T} = \mathbf{U}_R \mathbf{S}_R \tag{5.4.13}$$

and the loadings

$$\mathbf{P} = \mathbf{V}_R, \tag{5.4.14}$$

we can write the approximation $\widehat{\mathbf{X}}$ of $\mathbf{X}$ as

$$\widehat{\mathbf{X}} = \mathbf{T}\mathbf{P}^\top, \tag{5.4.15}$$

where the columns $\mathbf{t}_1, \ldots, \mathbf{t}_R$ of $\mathbf{T} \in \mathbb{R}^{N \times R}$ represent the scores and the columns $\mathbf{p}_1, \ldots, \mathbf{p}_R$ of $\mathbf{P} \in \mathbb{R}^{M \times R}$ represent the loadings. We can also express $\widehat{\mathbf{X}}$ directly as the sum of its components

$$\widehat{\mathbf{X}} = \mathbf{t}_1 \mathbf{p}_1^\top + \mathbf{t}_2 \mathbf{p}_2^\top + \cdots + \mathbf{t}_R \mathbf{p}_R^\top = \sum_{r=1}^{R} \mathbf{t}_r \mathbf{p}_r^\top, \tag{5.4.16}$$

Algorithm 5.1: NIPALS algorithm for PCA [355]

1 $\widetilde{\mathbf{X}}_0 = \mathbf{X}$
2 **for** $r = 1 \ldots R$ **do**
3 **repeat**
4 Select start value for $\mathbf{t}_r$ e.g. column in $\widetilde{\mathbf{X}}_{r-1}$ with highest sum of squares
  Improve estimate of $\mathbf{p}_r$
5 $\mathbf{p}_r^\top = (\mathbf{t}_r^\top \mathbf{t}_r)^{-1} \mathbf{t}_r{}^\top \widetilde{\mathbf{X}}_{r-1}$
  Scale $\mathbf{p}_r$ *to length one*
6 $\mathbf{p}_r = \frac{\mathbf{p}_r}{\|\mathbf{p}_r\|}$
  Improve estimate of $\mathbf{t}_r$
7 $\mathbf{t}_r = \widetilde{\mathbf{X}}_{r-1} \mathbf{p}_r (\mathbf{p}_r^\top \mathbf{p}_r)^{-1}$
  Improve estimate of eigenvalue λ_r
  $\lambda_{r,old} = \lambda_r$
8 $\lambda_r = \mathbf{t}_r^\top \mathbf{t}_r$
  until $(\lambda_r - \lambda_{r,old}) < \Theta$
  Check convergence
  Subtract the influence of the r-th component
9 $\widetilde{\mathbf{X}}_r = \widetilde{\mathbf{X}}_{r-1} - \mathbf{t}_r \mathbf{p}_r^\top$
end

as illustrated in Fig. 5.4 on the facing page. The contribution of the r-th component to $\widehat{\mathbf{X}}$ is represented by the rank-1 two-way array or *dyad* resulting from the outer product $\mathbf{t}_r \mathbf{p}_r^\top$ between $\mathbf{t}_r$ and $\mathbf{p}_r$.

One downside of this method is that even tough only the first few and largest R principal components will be used, all I singular values and corresponding singular values are extracted by the SVD. Especially for larger two-way arrays this may lead to unnecessary computational complexity in the numerical determination of the SVD, as we are only interested in a relatively small subset for the complete decomposition. Using the classic SVD computation algorithm by Golub-Kahan-Reinsch [74, 75], for example, this leads to $\mathcal{O}(M^3)$ floating points operations [76].

This can be avoided by determining the principal components iteratively with the *nonlinear iterative partial least squares (NIPALS)* algorithm proposed by Wold [355]. It uses the fact that the principal components' scores and loadings are orthogonal and extracts one principal component at a time. Algorithm 5.1 lists the NIPALS algorithm for the two-way feature array $\mathbf{X}$. For each principal component r the scores $\mathbf{t}_r$ and loadings $\mathbf{p}_r$ are estimated in the least square sense until convergence. As convergence criterion, the eigenvalue λ_r is used: in each iteration of the algorithm the difference between the eigenvalue in the current iteration λ_r and the eigenvalue in the previous iteration $\lambda_{r,old}$ is checked. If it is below a threshold Θ, the iteration is stopped and the scores and loadings of the current

iteration are considered to be representative of the r-th principal component, where e.g. Martens and Næs [197] suggest $\Theta = 10^{-4}$. The residual $\widetilde{\mathbf{X}}$ of the approximation of $\mathbf{X}$ after r principal components is then given by

$$\widetilde{\mathbf{X}}_r = \widetilde{\mathbf{X}}_{r-1} - \mathbf{t}_r \mathbf{p}_r^\top. \tag{5.4.17}$$

Note, that $\widetilde{\mathbf{X}}_r$ for each principal component r is equivalent to the residual and for all R principal components $\widetilde{\mathbf{X}}_R = \mathbf{E}_R$. This step of subtracting the influence of the r-th principal component from the approximation gained with the previous, larger the $r-1$-th principal component is called *deflating*. Thus in each step $\mathbf{X}$ is approximated better and only the variance of $\mathbf{X}$ not yet explained by the previous $r-1$ principal components is used for finding the r-th principal component.

Regression with the principal components After the desired R principal components with the corresponding scores $\mathbf{T}$ and loadings $\mathbf{P}$ have been determined, the visual quality vector $\mathbf{y}$ can be regressed upon the approximated two-way feature array $\widehat{\mathbf{X}}$. Thus we gain a prediction model for $\mathbf{y}$ using only the first R principal components under the assumption that these principal components describe the structure of the data well enough and that the $M-R$ components not included in the model describe mainly unsystematic variation caused by noise. Note, however, that for $R = M$ the PCR provides the same model as the MLR. As we are interested on the influence of the latent variables presented by the principal components, we express $\mathbf{y}$ as

$$\mathbf{y} = \mathbf{T}\mathbf{q}, \tag{5.4.18}$$

where $\mathbf{q} \in \mathbb{R}^{R\times 1}$ are the weights with respect to the scores $\mathbf{T}$. Hence the weight vector $\mathbf{q}$ represents the influence of the principal components on the visual quality with q_1 corresponding to the first principal component and q_R to the last principal component in our approximation $\widehat{\mathbf{X}}$ of $\mathbf{X}$. Applying MLR and therefore using the least square estimate $\widehat{\mathbf{q}}$ for $\mathbf{q}$, we get

$$\widehat{\mathbf{q}} = (\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{y}, \tag{5.4.19}$$

where we used that the columns of $\mathbf{T}$ are orthonormal and therefore the inverse of $\mathbf{T}^\top\mathbf{T}$ exists. The predicted visual quality vector $\widehat{\mathbf{y}}$ for the N sequences in the calibration set $\mathcal{S}_C$ is then given as

$$\widehat{\mathbf{y}} = \mathbf{T}\widehat{\mathbf{q}}, \tag{5.4.20}$$

and the true visual quality vector $\mathbf{y}$ can be written as

$$\mathbf{y} = \mathbf{T}\widehat{\mathbf{q}} + \mathbf{e}, \tag{5.4.21}$$

where $\mathbf{e}$ is the residual. With $\mathbf{T}$ as $\mathbf{T} = \widehat{\mathbf{X}}\mathbf{P}$ from (5.4.15), we can rewrite (5.4.20) as

$$\widehat{\mathbf{y}} = \widehat{\mathbf{X}}\mathbf{P}\widehat{\mathbf{q}}, \tag{5.4.22}$$

and get the weights $\widehat{\mathbf{b}} \in \mathbb{R}^{M\times 1}$ relating the two-way feature array $\widehat{\mathbf{X}}$ directly to the visual quality vector $\mathbf{y}$ as

$$\widehat{\mathbf{b}} = \mathbf{P}\widehat{\mathbf{q}}, \tag{5.4.23}$$

that allows us to write the prediction for the PCR in the same form as for the MLR in (5.2.8):

$$\widehat{\mathbf{y}} = \widehat{\mathbf{X}}\widehat{\mathbf{b}}. \tag{5.4.24}$$

In order to predict the visual quality $\widehat{y}_u$ of a unknown video sequence $\underline{\mathbf{S}}_U \notin \mathcal{S}_C$ and the corresponding feature vector $\mathbf{x}_u$, we can directly use the weights $\widehat{\mathbf{b}}$ or we first obtain the scores $\mathbf{t}_u$ for $\mathbf{x}_u$ with $\mathbf{t}_u = \mathbf{x}_u\mathbf{P}$ and then use the weights $\widehat{\mathbf{q}}$

$$\widehat{y}_u = \mathbf{x}_u\widehat{\mathbf{b}} + \widehat{b}_0 = \mathbf{x}_u\mathbf{P}\widehat{\mathbf{q}} + \widehat{b}_0 = \mathbf{t}_u\widehat{\mathbf{q}} + \widehat{b}_0. \tag{5.4.25}$$

Note, that we assume that $\mathbf{x}_u$ has been preprocessed and $\widehat{b}_0 = \overline{y}$.The prediction weights $\widehat{\mathbf{b}}$ without preprocessing are discussed in Section 5.2 on the MLR. For more information about PCA and PCR, I refer to Jolliffe [128].

PCR addressed the issue that not all features influence the variation within the two-way feature array $\mathbf{X}$ equally: the principal components reflecting the latent variables of $\mathbf{X}$ are used to gain a more stable prediction model by only including the most important components in the model. But we did not consider how relevant these principal components are for the subjective quality vector $\mathbf{y}$ and some principal components may not have any influence on $\mathbf{y}$ at all. Our goal, however, is a prediction model for the visual quality of unknown video sequences $\underline{\mathbf{S}}_U$ and the model should therefore not only describe the hidden structure of the features well, but should also provide a good ability to predict the visual quality $\widehat{y}_u$ of these unknown sequences. Hence the components in our component model must not only provide a good description of $\mathbf{X}$, but also of its relationship with $\mathbf{y}$.

5.5. Partial least squares regression (PLSR)

Partial least squares (PLS) regression is a bilinear method that addresses the major shortcoming of the PCR by using the subjective quality vector $\mathbf{y}$ actively during the decomposition of $\mathbf{X}$ into components. PLS in its basic form was first presented by Herman Wold in [357] based on his previous concepts in [356]. His son Svante Wold introduced PLS later into the chemometrics community in [358, 359], providing an interpretation of PLS in the context of chemometrics and the corresponding definitions that are also used in this thesis. PLS follows the paradigm of *no predictor without interpretation, no interpretation without predictive ability* [197], reflecting the idea that the components should provide a good description of the latent variables in the two-way feature array $\mathbf{X}$ and at the same time should also be able to predict the visual quality $\mathbf{y}$ well. Svante Wold suggests therefore in [365] that *projection on latent structures* may be a better definition of PLS.

PLS is also called *PLS1* for a univariate dependent variable i.e. $\mathbf{y}$ is a one-way array , and for a multivariate dependent variable i.e. $\mathbf{Y}$ is a two-way array, PLS is called *PLS2*.

In this thesis, the focus is on the univariate PLS1 as the visual quality $\mathbf{y}$ is represented by vector and in order to keep a consistent notation with the previous sections PLS1 will be referred to as PLSR from now on.

General idea behind PLS Using the two-way feature array $\mathbf{X} \in \mathbb{R}^{N \times M}$ and the visual quality vector $\mathbf{y} \in \mathbb{R}^{N \times 1}$, PLS aims to maximaise the covariance between $\mathbf{y}$ and the scores $\mathbf{t}_r \in \mathbb{R}^{N \times 1}$ with respect to weights $\mathbf{w}_r \in \mathbb{R}^{M \times 1}$ for $r1 \ldots R$ components. For the first PLS component and corresponding score $\mathbf{t}_1$ this can be written as

$$\max_{\|\mathbf{w}_1\|=1} \left[\text{cov}(\mathbf{t}_1, \mathbf{y}) | \mathbf{t}_1 = \mathbf{X}\mathbf{w}_1 \right], \tag{5.5.1}$$

where $\mathbf{w}_1$ is restricted to length one in order to avoid that the covariance becomes arbitrarily large. By using that $\mathbf{X}$ is mean centred and omitting the correction for the degrees of freedom in the covariance, we can rewrite (5.5.1) as

$$\max_{\|\mathbf{w}_1\|=1} \left[\mathbf{y}^\top \mathbf{t}_1 | \mathbf{t}_1 = \mathbf{X}\mathbf{w}_1 \right], \tag{5.5.2}$$

that can be further rewritten as

$$\max_{\|\mathbf{w}_1\|=1} \left[\mathbf{y}^\top \mathbf{X}\mathbf{w}_1 \right]. \tag{5.5.3}$$

With the substitution $\mathbf{z} = \mathbf{X}^\top \mathbf{y}$ we arrive at

$$\max_{\|\mathbf{w}_1\|=1} \left[\mathbf{z}^\top \mathbf{w}_1 \right], \tag{5.5.4}$$

that is maximised if and only if [91]

$$\mathbf{w}_1 = \frac{\mathbf{z}}{\|\mathbf{z}\|} = \frac{\mathbf{X}^\top \mathbf{y}}{\|\mathbf{X}^\top \mathbf{y}\|}. \tag{5.5.5}$$

After finding the first PLS component and corresponding score $\mathbf{t}_1$, both $\mathbf{X}$ and $\mathbf{y}$ are deflated be removing the structural part modelled by the first component. The second PLS component is then determined from the deflated $\mathbf{X}$, similar to the NIPALS algorithm for PCA.

Considering the maximisation criterion of the covariance between $\mathbf{t}$ and $\mathbf{y}$, that maximises both the variance within $\mathbf{t}$ but also the correlation between $\mathbf{t}$ and $\mathbf{y}$, the PLS components can be characterised as on the one hand having a large enough variation in $\mathbf{t}$ and thus ensuring that noise is not modelled, and on the other hand of providing a correlation between $\mathbf{t}$ and $\mathbf{y}$ high enough so that $\mathbf{t}$ has predictive relevance for $\mathbf{t}$ [298].

Three of the most common PLSR algorithms will be discussed briefly: the original orthogonal NIPALS PLS1 algorithm as proposed by Wold et al. [358], the non-orthogonal version of the NIPALS PLS1 algorithm as proposed by Martens and Næs [196], and lastly the SIMPLS algorithm proposed by de Jong [129]. Other PLSR algorithms not discussed here include the Kernel PLS by Lindgren et al. [182] and the PLS extension O-PLS by Trygg and Wold [323], and I refer to [182, 323] for further information about these algorithms.

Algorithm 5.2: Orthogonal NIPALS PLS1 algorithm [358]

$\widetilde{\mathbf{X}}_0 = \mathbf{X}$
$\widetilde{\mathbf{y}}_0 = \mathbf{y}$
for $r = 1 ... R$ **do**
  Find weights $\mathbf{w}_r$ *scaled to length one by using remaining variability*
    $\mathbf{w}_r = \frac{\widetilde{\mathbf{X}}_{r-1}^\top \widetilde{\mathbf{y}}_{r-1}}{\|\widetilde{\mathbf{X}}_{\mathbf{r-1}}^\top \widetilde{\mathbf{y}}_{\mathbf{r-1}}\|}$
  Estimate scores $\mathbf{t}_r$
    $\mathbf{t}_r = \widetilde{\mathbf{X}}_{r-1}\mathbf{w}_r$
  Estimate loadings $\mathbf{p}_r$
    $\mathbf{p}_r = \widetilde{\mathbf{X}}_{r-1}^\top \mathbf{t}_r (\mathbf{t}_r^\top \mathbf{t}_r)^{-1}$
  Estimate weights $\widehat{q}_r$
    $\widehat{q}_r = \widetilde{\mathbf{y}}_{r-1}^\top \mathbf{t}_r (\mathbf{t}_r^\top \mathbf{t}_r)^{-1}$
  Subtract the influence of the r-th component from $\widetilde{\mathbf{X}}_{r-1}$ *and* $\widetilde{\mathbf{y}}_{r-1}$
    $\widetilde{\mathbf{X}}_r = \widetilde{\mathbf{X}}_{r-1} - \mathbf{t}_r \mathbf{p}_r^\top$
    $\widetilde{\mathbf{y}}_r = \widetilde{\mathbf{y}}_{r-1} - \mathbf{t}_r \widehat{q}_r$
end

Orthogonal NIPALS PLS1 algorithm The original orthogonal NIPALS PLS1 algorithm proposed by Wold et al. [358] described in Algorithm 5.2 provides orthogonal scores $\mathbf{t}_r$ and weights $\mathbf{w}_r$, but the loadings are $\mathbf{p}_r$ generally not orthogonal. Similar to the PCR, the aim is to arrive at an R component model with $R < M$, where the assumption is that the $M - R$ components not included in the model describe mainly unsystematic variation. Note, that for $R = M$ the model gained with PLS is the same model as gained with MLR.

In each iteration, the weights $\mathbf{w}_r$ that maximise the covariance between visual quality $\mathbf{y}$ and scores $\mathbf{t}_r$ are determined according to (5.5.5) by considering the remaining variability from the previous iteration $r-1$ in $\widetilde{\mathbf{X}}_{r-1}$ and $\widetilde{\mathbf{y}}_{r-1}$, where $\widetilde{\mathbf{X}}_r$ and $\widetilde{\mathbf{y}}_r$ describe the residual of unexplained variability of $\mathbf{X}$ and $\mathbf{y}$ after the r-th iteration. Based on these weights $\mathbf{w}_r$, the scores $\mathbf{t}_r$, the loadings $\mathbf{p}_r$ and the regression weights $\widehat{q}_r$ are estimated in the least squares sense, before deflating $\widetilde{\mathbf{X}}_{r-1}$ and $\widetilde{\mathbf{y}}_{r-1}$ using the new $\mathbf{t}_r$, $\mathbf{p}_r$ and $\widehat{q}_r$. Note, that although part of the algorithm, it is not necessary to deflate $\widetilde{\mathbf{y}}_r$, as the deflation of $\widetilde{\mathbf{X}}_r$ already ensures that the remaining data in $\widetilde{\mathbf{X}}_r$ for the next iteration is orthogonal to the part of $\widetilde{\mathbf{y}}_r$ described in the current iteration r [49, 134]. Unlike for the PCR, the estimation of the regression weights $\widehat{q}_r$ is not a separate step done after all scores and loadings had been determined with PCA, but rather an integral part of the algorithm.

After R components have been determined, the columns $\mathbf{t}_1, ..., \mathbf{t}_R$ of $\mathbf{T} \in \mathbb{R}^{N \times R}$ represent the scores, the columns $\mathbf{p}_1, ..., \mathbf{p}_R$ of $\mathbf{P} \in \mathbb{R}^{M \times R}$ represent the loadings, the columns $\mathbf{w}_1, ..., \mathbf{w}_R$ of $\mathbf{W} \in \mathbb{R}^{M \times R}$ represent the weights and the rows $\widehat{q}_1, ... \widehat{q}_R$ of $\widehat{\mathbf{q}} \in \mathbb{R}^{R \times 1}$ represent the regression weights for each of the r components. We can then write the scores

as

$$\mathbf{T} = \mathbf{X}\mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1} \tag{5.5.6}$$

and the weight matrix $\widehat{\mathbf{V}}$ discussed in Section 5.3 for this algorithm is therefore given as $\widehat{\mathbf{V}} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}$ [84]. It can be shown that for given orthogonal scores $\mathbf{T}$ the corresponding loadings $\mathbf{P} = (\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{X}$ then solve the minimisation problem

$$\min_{\mathbf{T}} \|\mathbf{X} - \mathbf{T}\mathbf{P}^\top\|^2, \tag{5.5.7}$$

where the orthogonality of the scores $\mathbf{t}_r$ in $\mathbf{T}$ is used. This, however, is not the case for general $\mathbf{T}$ and $\mathbf{P}$, but only for the $\mathbf{P}$ corresponding to the given $\mathbf{T}$, and therefore partial least squares is called *partial* least squares [91, 299]. The approximation of the two-way feature array $\mathbf{X}$ after R components can then be written as

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E}, \tag{5.5.8}$$

and the true visual quality $\mathbf{y}$ as

$$\mathbf{y} = \mathbf{T}\widehat{\mathbf{q}} + \mathbf{e}, \tag{5.5.9}$$

where $\mathbf{E}$ and $\mathbf{e}$ are the residual of $\mathbf{X}$ and $\mathbf{y}$, respectively. The prediction $\widehat{\mathbf{y}}$ of the visual quality $\mathbf{y}$ is then given by

$$\widehat{\mathbf{y}} = \mathbf{T}\widehat{\mathbf{q}}, \tag{5.5.10}$$

and with $\mathbf{T} = \mathbf{X}\mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}$ from (5.5.6) we can rewrite this to

$$\widehat{\mathbf{y}} = \mathbf{X}\mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-\mathbf{1}}\widehat{\mathbf{q}}. \tag{5.5.11}$$

Using (5.5.11), the weights $\widehat{\mathbf{b}} \in \mathbb{R}^{M\times 1}$ relating the two-way feature array $\mathbf{X}$ directly to the visual quality vector $\mathbf{y}$ can be expressed as

$$\widehat{\mathbf{b}} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}\widehat{\mathbf{q}}. \tag{5.5.12}$$

The prediction of the the visual quality $\widehat{y}_u$ for a unknown video sequence $\underline{\mathbf{S}}_U \notin \mathcal{S}_C$ and the corresponding feature vector $\mathbf{x}_u$ is then the same as for the PCR or MLR as described in the previous sections.

Non-orthogonal PLS1 algorithm The non-orthogonal NIPALS PLS1 algorithm proposed by Martens and Næs [196] described in Algorithm 5.3 on the next page is an extension of the original orthogonal PLS1 algorithm described in Algorithm 5.2 on the preceding page.

In difference to the orthogonal algorithm, we no longer need to compute loadings $\mathbf{p}_r$, but only use the weights $\mathbf{w}_r$. One consequence of this is, however, that the scores $\mathbf{t}_r$ are no longer orthogonal and therefore we can no longer perform the regression of the scores $\mathbf{t}_r$ onto the visual quality $\mathbf{y}$ for each of the r components individually, but rather need to

Algorithm 5.3: Non-orthogonal NIPALS PLS1 algorithm [196]

$\widetilde{\mathbf{X}}_0 = \mathbf{X}$
$\widetilde{\mathbf{y}}_0 = \mathbf{y}$
for $r = 1 \ldots R$ **do**
  Find weights $\mathbf{w}_r$ *scaled to length one by using remaining variability*
    $\mathbf{w}_r = \dfrac{\widetilde{\mathbf{X}}_{r-1}^\top \widetilde{\mathbf{y}}_{r-1}}{\|\widetilde{\mathbf{X}}_{\mathbf{r-1}}^\top \widetilde{\mathbf{y}}_{\mathbf{r-1}}\|}$
  Estimate scores $\mathbf{t}_r$
    $\mathbf{t}_r = \widetilde{\mathbf{X}}_{r-1}\mathbf{w}_r$
  Estimate weights $\widehat{\mathbf{q}}$
    $\widehat{\mathbf{q}} = (\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\widetilde{\mathbf{y}}_{r-1}$
  Subtract the influence of the r*-th component from* $\widetilde{\mathbf{X}}_{r-1}$ *and* $\widetilde{\mathbf{y}}_{r-1}$
    $\widetilde{\mathbf{X}}_r = \widetilde{\mathbf{X}}_{r-1} - \mathbf{t}_r\mathbf{w}_r^\top$
    $\widetilde{\mathbf{y}}_r = \widetilde{\mathbf{y}}_{r-1} - \mathbf{T}\widehat{\mathbf{q}}$
end

perform a simultaneous regression of all r scores determined so far on $\mathbf{y}$ in each iteration. Hence we estimate in the r-th iteration the regression weights $\widehat{\mathbf{q}}_r \in \mathbb{R}^{r\times 1}$ in the least square sense with $\mathbf{T} \in \mathbb{R}^{N\times r}$, where the columns $\mathbf{t}_1 \ldots \mathbf{t}_r$ represent the scores obtained in all iterations including the current r-th iteration. If necessary, a loading matrix $\mathbf{P}$ and the loadings $\mathbf{p}_r$ can be determined by applying MLR on $\mathbf{X}$ and $\mathbf{T}$

$$\mathbf{P} = (\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{X}. \tag{5.5.13}$$

With the same definitions as for the orthogonal algorithm, we can then write the scores as

$$\mathbf{T} = \mathbf{XW} \tag{5.5.14}$$

and the weight matrix $\widehat{\mathbf{V}}$ discussed in Section 5.3 for this algorithm is therefore given as $\widehat{\mathbf{V}} = \mathbf{W}$ [197]. For non-orthogonal scores $\mathbf{T}$, it can be shown that $\mathbf{T} = \mathbf{XW}$ solves the minimisation problem

$$\min_{\mathbf{T}} \|\mathbf{X} - \mathbf{TW}^\top\|^2, \tag{5.5.15}$$

where $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$ is used. Similar to orthogonal scores, however, $\mathbf{T} = \mathbf{XW}$ does not solve the minimisation problem for $\mathbf{T}$ and $\mathbf{W}$ in general, because $\mathbf{W}$ is constrained to define the direction of maximum covariance with $\mathbf{y}$ [299]. The approximation of the two-way feature array $\mathbf{X}$ after R components can then be written as

$$\mathbf{X} = \mathbf{TW}^\top + \mathbf{E}, \tag{5.5.16}$$

and the true visual quality $\mathbf{y}$ as before as

$$\mathbf{y} = \mathbf{T}\widehat{\mathbf{q}} + \mathbf{e}, \tag{5.5.17}$$

where $\mathbf{E}$ and $\mathbf{e}$ are the residual of $\mathbf{X}$ and $\mathbf{y}$, respectively. With $\widehat{\mathbf{y}} = \mathbf{T}\widehat{\mathbf{q}}$ from (5.5.10) and $\mathbf{T} = \mathbf{XW}$ from (5.5.14) the prediction $\widehat{\mathbf{y}}$ of the visual quality $\mathbf{y}$ is then given by

$$\widehat{\mathbf{y}} = \mathbf{XW}\widehat{\mathbf{q}}. \tag{5.5.18}$$

Using (5.5.18), the weights $\widehat{\mathbf{b}} \in \mathbb{R}^{M\times 1}$ relating the two-way feature array $\mathbf{X}$ directly to the visual quality vector $\mathbf{y}$ can be expressed as

$$\widehat{\mathbf{b}} = \mathbf{W}\widehat{\mathbf{q}}. \tag{5.5.19}$$

The prediction of the the visual quality for a unknown video sequences is then the same as described in the previous sections.

Although the loadings weights $\mathbf{W}$ are the same in both the orthogonal and non-orthogonal algorithm, scores $\mathbf{T}$ and regression weighs $\widehat{\mathbf{q}}$ will have different values. The solution in terms of the model fitted to the visual quality $\mathbf{y}$ and the prediction of the visual quality of unknown sequences y_u, however, is identical for both algorithms [84].

SIMPLS algorithm The SIMPLS algorithm proposed by de Jong [129] described in Algorithm 5.4 on the next page provides a different approach to the PLS regression. It derives the PLS components directly from the two-way feature array $\mathbf{X}$ and does not need to deflate either $\mathbf{X}$ or $\mathbf{y}$ as in the two PLS algorithms discussed so far. Additionally, the scores $\mathbf{t}_r$ are orthogonal. The algorithm utilises the property, that the maximisation of the covariance for $\mathbf{X}^\top\mathbf{Y}$ can be solved with the first left singular vector of $\mathbf{X}^\top\mathbf{Y}$. Although this degenerates for the PLS1 case with a vector $\mathbf{y}$ and therefore $\mathbf{X}^\top\mathbf{y}$ as seen in Algorithm 5.4 on the facing page [129], this step was maintained in the listing as it is an integral part of the SIMPLS algorithm. Scores $\mathbf{t}_r$, loadings $\mathbf{p}_r$, regression weights $\widehat{q}_r$ are then estimated in the least squares sense. Instead of deflating $\mathbf{X}$ or $\mathbf{y}$, $\mathbf{s} = \mathbf{X}^\top\mathbf{y}$ is deflated in each iteration by projecting $\mathbf{s}$ on a subspace orthogonal to the loadings $\mathbf{P}$. Note, that the weights $\mathbf{w}_r$ from the previous NIPALS algorithms have been replaced by the weights $\mathbf{r}_r$ to reflect the different meaning of the weights in the notation.

With the same definitions as for the NIPALS algorithms and additionally with the columns $\mathbf{r}_1, \ldots, \mathbf{r}_R$ of $\mathbf{R} \in \mathbb{R}^{M\times R}$ represent the weights, we can then write the scores as

$$\mathbf{T} = \mathbf{XR} \tag{5.5.20}$$

and the weight matrix $\widehat{\mathbf{V}}$ discussed in Section 5.3 for this algorithm is therefore given as $\widehat{\mathbf{V}} = \mathbf{R}$. By replacing $\mathbf{W}$ with $\mathbf{R}$, we can express the approximation of the two-way feature array $\mathbf{X}$, the true visual quality $\mathbf{y}$ and the prediction $\widehat{\mathbf{y}}$ of the visual quality $\mathbf{y}$ after R components as for the non-orthogonal NIPALS algorithm in (5.5.16), (5.5.17) and (5.5.18), respectively.

The weights $\widehat{\mathbf{b}} \in \mathbb{R}^{M\times 1}$ relating the two-way feature array $\mathbf{X}$ directly to the visual quality vector $\mathbf{y}$ can be expressed as

$$\widehat{\mathbf{b}} = \mathbf{R}\widehat{\mathbf{q}}. \tag{5.5.21}$$

Algorithm 5.4: SIMPLS PLS1 algorithm [129]

1 $\mathbf{s}_0 = \mathbf{X}^\top \mathbf{y}$
2 **for** $r = 1 \ldots R$ **do**
  Initialise weights $\widehat{\mathbf{q}}_r$
3 $\widehat{\mathbf{q}}_r$ = first left singular vector of $\mathbf{s}_{r-1}^\top \mathbf{s}_{r-1}$
  here $\widehat{\mathbf{q}}_r = \widehat{q}_r = 1$
  Find weights $\mathbf{r}_r$
4 $\mathbf{r}_r = \mathbf{s}_{r-1}\widehat{\mathbf{q}}_r$
  Estimate scores $\mathbf{t}_r$
5 $\mathbf{t}_r = \mathbf{X}\mathbf{r}_r$
  Normalise scores $\mathbf{t}_r$
6 $\mathbf{t}_r = \frac{\mathbf{t}_r}{\|\mathbf{t}_r\|}$
  Adapt weights $\mathbf{r}_r$
7 $\mathbf{r}_r = \frac{\mathbf{r}_r}{\|\mathbf{t}_r\|}$
  Estimate loadings $\mathbf{p}_r$
8 $\mathbf{p}_r = \mathbf{X}^\top \mathbf{t}_r$
  Estimate weights $\widehat{q}$
9 $\widehat{q}_r = \mathbf{y}^\top \mathbf{t}_r$
  Deflate $\mathbf{s}$ *with respect to loadings*
10 $\mathbf{s}_r = \mathbf{s}_{r-1} - \mathbf{P}(\mathbf{P}^\top \mathbf{P})^{-1}\mathbf{P}^\top \mathbf{s}$
end

The prediction of the the visual quality for a unknown video sequences is then the same as described in the previous sections.

For the univariate case i.e. PLS1, SIMPLS is equivalent to the NIPALS algorithm, for the multivariate case i.e. PLS2, however, there is a slight difference in the scores and weights [129]. As SIMPLS only deflates the comparably small vector $\mathbf{s}_r$ in each iteration, whereas NIPALS deflates in each iteration the potentially large two-way array $\mathbf{X}$ and thus needs more matrix operations, it can be assumed that SIMPLS is computationally less complex. In the context of chemometrics, empirical results for commonly used data sets show that SIMPLS is in most cases more than twice as fast as (orthogonal) NIPALS [201].

One question not addressed so far, is if the inclusion of the subjective quality vector $\mathbf{y}$ in the component model building process really improves the prediction performance of models built with PLSR compared to those with PCR. Although I address this in the context of video quality metric design in the performance comparison of the different data analysis methods in Chapter 9, there have been so far some contributions discussing this issue: in [67], Frank and Friedman show that for simulated data PCR and PLSR provide similar results with a slight advantage for PLSR, in [84], Helland also shows that for typical data in chemometrics, PCR and PLSR deliver similar results if only the most relevant

principal components are selected for the PCR, in [310], Stoica and Söderström provide a theoretical explanation for the similarity of the prediction results for PCR and PLSR and in [85], Helland and Almoy argue that PLSR should be preferred as it tends to exclude components that are of intermediate irrelevance better than PCR.

For more information about PLS in general, but also for a description of the multivariate PLS2, I refer to Martens and Næs [197], and for a statistical analysis of PLS to Höskuldsson [91]. Information about the history behind PLS, is available in Wold [361] and Martens [198].

Non-linear data One implicit assumption in the application of linear two-way data analysis methods so far has been that the relationship between the features in the two-way feature array $\mathbf{X}$ and the visual quality vector $\mathbf{y}$ is linear. It may, however, very well be that some features have a non-linear relationship to the visual quality, especially considering the non-linear nature of some properties of human perception.

Bilinear data analysis methods are generally able to model a certain amount of non-linear structures in $\mathbf{X}$ at the cost of an significantly increased number of components as long as there is not only a non-linear relationship between $\mathbf{y}$ and $\mathbf{X}$, but also between the individual features in $\mathbf{X}$ [197]. The increased number of components can be explained as follows: in each iteration the scores and weights for the current r-th component are determined from the residuals $\widetilde{\mathbf{X}}_{r-1}$ and $\widetilde{\mathbf{y}}_{r-1}$ unexplained by the previous $r-1$ components. Considering that this residual is likely to still contain a non-linearity between the residuals $\widetilde{\mathbf{X}}_{r-1}$ and $\widetilde{\mathbf{y}}_{r-1}$, the current r-th component then explains part of this non-linearity by a linear least squares approximation. Thus each additional component provides a different linear approximation of the remaining non-linearity in each iteration and the additionally needed components can be considered as a successive linear approximation of the non-linearity.

If this inherent tolerance of non-linear structures is not sufficient, either the features in $\mathbf{X}$ can be mapped by a non-linear projection in order to achieve again a linear relationship between $\mathbf{y}$ and $\mathbf{X}$ as input for the data analysis methods, or the least squares estimation of the regression weights $\mathbf{q}$ with respect to the scores $\mathbf{t}$ is replaced in the NIPALS algorithm by a non-linear regression. Both options, however, necessitate that we have some knowledge about the non-linearity, which is usually not readily available. Although non-linear PLSR modifications have been suggested in literature for some time e.g. Quadrilinear PLSR by Wold et al. [364] and Höskuldsson [92], these modification have so far not been widely studied or adopted. For more information about the handling of non-linear data in the context of data analysis and an overview of non-linear PLSR approaches, I refer to Martens and Næs [197] and the recent survey by Rosipal [273] and references therein.

6. Multi-way Data Analysis

In the previous chapter, temporal pooling was applied to generate a two-way representation of the three-way data in order to be able to apply two-way data analysis methods. But the pooling does not necessarily reflect the temporal variation within the three-way feature array $\underline{\mathbf{X}}$ sufficiently, as depending on the choice of the pooling function, only certain statistical properties of the variation are retained in the resulting two-way feature array $\mathbf{X}$.

Multi-way data analysis therefore aims to avoid any pooling and applies the methods directly to the multi-way data, in our case represented by the the three-way feature array $\underline{\mathbf{X}}$. We can either apply two-way data analysis methods on the multi-way data resulting in two-way models or use directly multi-way data analysis methods, leading to multi-way models.

In this section, I therefore first discuss two methods to perform two-way data analysis on three-way data without previous pooling. Following this intermediate step between purely two-way data analysis and multi-way data analysis, I briefly review the extension of the concepts of two-way data analysis to multi-way data analysis with multi-way component models allowing us to describe three-way data, before introducing the three-way data analysis method used in this thesis, the trilinear PLSR that is the three-way extension of the two-way PLSR discussed in Section 5.5. As the feature array $\underline{\mathbf{X}}$ is a three-way array, the focus is on three-way data analysis, but most of the discussed methods can be extended straightforwardly to n-way arrays.

6.1. Two-way data analysis with three-way data

Two-way data analysis with three-way data can be considered as an intermediate step between two-way and multi-way data analysis, as we apply two-way data analysis methods on the three-way data. One advantage of this approach is that we can use all two-way data analysis methods, while still maintaining the multi-way nature of the data. The disadvantage, however, is clearly that two-way models may not necessarily be the best way to describe three-way data. In this section, I discuss two such methods, *unfolding* and the *bilinear 2D-PCR*.

6.1.1. Unfolding and bilinear methods

Unfolding along the temporal mode is obviously the easiest option to handle the three-way data without temporal pooling, as we only have to rearrange the frontal slices $\mathbf{X}_t$ of $\underline{\mathbf{X}}$.

Once the data is unfolded, two-way data analysis methods as described in Chapter 5 can be applied to the data in order to build prediction models. We first perform the mode-1 unfolding of the three-way feature array $\underline{\mathbf{X}}$ into the two-way feature array $\mathbf{X}_{N\times MT} \in \mathbb{R}^{N\times MT}$ by concatenating the frontal slices $\mathbf{X}_t \in \mathbb{R}^{N\times M}$ of $\underline{\mathbf{X}}$ in ascending order. Then we can apply bilinear MLR, PCR or PLSR on $\mathbf{X}_{N\times MT}$ resulting in a weight vector $\widehat{\mathbf{b}} \in \mathbb{R}^{MT\times 1}$ as described in Algorithm 6.1 on the next page.

For the prediction of the visual quality $\widehat{y}_u$ of a unknown video sequence $\underline{\mathbf{S}}_U \notin \mathcal{S}_C$, the mode-1 unfolding is performed on the corresponding horizontal feature slice $\mathbf{X}_u \in \mathbb{R}^{M\times T}$, resulting in a feature vector $\mathbf{x}_u \in \mathbb{R}^{1\times MT}$. The visual quality can then be predicted using $\mathbf{x}_u$ and $\widehat{\mathbf{b}}$ with

$$\widehat{y}_u = \mathbf{x}_u\widehat{\mathbf{b}}, \tag{6.1.1}$$

as discussed in the previous sections for the two-way data. Note, that once again we assume that both $\mathbf{X}_{N\times MT}$ and $\mathbf{X}_u$ have been preprocessed similar to the centring and scaling of the pooled two-way array $\mathbf{X}$. The prediction weights $\widehat{\mathbf{b}}$ for the quality prediction without preprocessing can be determined as described in Section 5.2.

Alternatively, the weight vector $\widehat{\mathbf{b}}$ for the unfolded data can be expressed as a two-way weight array $\widehat{\mathbf{B}} \in \mathbb{R}^{M\times T}$ by rearranging the elements of $\widehat{\mathbf{b}}$ as

$$\widehat{\mathbf{B}} = \begin{bmatrix} \widehat{b}_1 & \widehat{b}_{M+1} & \cdots & \widehat{b}_{M(T-1)+1} \\ \widehat{b}_2 & \widehat{b}_{M+2} & \cdots & \widehat{b}_{M(T-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{b}_M & \widehat{b}_{M2} & \cdots & \widehat{b}_{MT} \end{bmatrix}, \tag{6.1.2}$$

where the t-th column $\widehat{\mathbf{b}}_t$ of $\widehat{\mathbf{B}}$ represents the M weights for the t-th frame. The prediction for the visual quality $\widehat{y}_u$ can then be written as the inner product between $\mathbf{X}_u$ and $\widehat{\mathbf{B}}$ as

$$\widehat{y}_u = \langle \mathbf{X}_u, \widehat{\mathbf{B}} \rangle. \tag{6.1.3}$$

Similar to the weights $\widehat{\mathbf{b}}$ corresponding to $\widehat{\mathbf{b}}$ for the visual quality prediction without preprocessing, we can also determine a two-way weight array $\widehat{\mathbf{B}}$ and offset $\widehat{b}_0$ incorporating the scaling and centring of the feature slice $\mathbf{X}_u$ of the unknown video sequence $\underline{\mathbf{S}}_u$. Based on

$$\widehat{y}_u = \langle \mathbf{W}(\mathbf{X}_u - \mathbf{M}), \widehat{\mathbf{B}} \rangle + \overline{y}, \tag{6.1.4}$$

where, $\mathbf{M}$ is the matrix containing the column average from the mode-1 centring of all horizontal slices according to (4.2.9), $\mathbf{W}$ is the mode-2 scaling matrix for the individual features according to (4.2.14) and $\overline{y}$, the new two-way weight array $\widehat{\mathbf{B}}$ is then given by

$$\widehat{\mathbf{B}} = \mathbf{W}\widehat{\mathbf{B}} \tag{6.1.5a}$$

and

$$\widehat{b}_0 = \overline{y} - \langle \mathbf{WM}, \widehat{\mathbf{B}} \rangle. \tag{6.1.5b}$$

Algorithm 6.1: Unfolding and bilinear methods

Unfold $\underline{\mathbf{X}}$ *along first mode by rearranging frontal slices* $\mathbf{X}_t$ *of* $\underline{\mathbf{X}}$

$\mathbf{X}_{N\times MT} = \begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_t & \cdots & \mathbf{X}_T\end{bmatrix}$

Perform two-way analysis on $\mathbf{X}_{N\times MT}$

$\widehat{\mathbf{b}} \leftarrow \mathrm{MLR}(\mathbf{y}, \mathbf{X}_{N\times MT})$

or

$\widehat{\mathbf{b}} \leftarrow \mathrm{PCR}(\mathbf{y}, \mathbf{X}_{N\times MT})$

or

$\widehat{\mathbf{b}} \leftarrow \mathrm{PLSR}(\mathbf{y}, \mathbf{X}_{N\times MT})$

Optional for prediction without unfolding

Rearrange $\widehat{\mathbf{b}}$ into $\widehat{\mathbf{B}}$

The visual quality for the unknown video sequence $\mathcal{S}_U$ without preprocessing of its feature slice $\mathbf{X}_u$ can then be expressed as

$$\widehat{y}_u = \langle \mathbf{X}_u, \widehat{\mathbf{B}} \rangle + \widehat{b}_0. \tag{6.1.6}$$

Unfolding and the subsequent application of two-way data analysis methods, however, has some drawbacks: firstly, the unfolding and therefore reshaping of the three-way data into two-way data breaks the inherent structure and possible correlations between the different modes, therefore removing redundancies and dependencies that could be leveraged for a more compact and useful description of the data [188]. Secondly, the loadings $\mathbf{P}$ in the two-way model for the unfolded three-way data have to simultaneously describe the influence of variables in two different modes and are therefore not necessarily the best description of each of the individual modes. Also this makes the interpretation of the model more difficult. Lastly, the two-way array created by the unfolding of the three-way data is significantly larger than the two-array of the temporally pooled features in the previous sections: usually the number of frames T is larger than the number of features M by at least a factor of 10^2–10^3, hence $T \gg M$ and therefore also $MT \gg M$. Depending on the applied two-way data analysis method, this can lead to increased computational complexity for some methods.

Bro shows in [28] that unfolding leads to *less robust, less interpretable, less predictive* and *non-parsimonious* models for common applications and corresponding data in chemometrics. He suggests that generally the better data can be approximated by a multi-way structure and the noiser the data is, the more beneficial it is to use multi-way data analysis methods instead of unfolding of the three-way data and using two-way data analysis methods. Considering that video has an inherent three-way structure due the temporal variation of the features, it is therefore likely that unfolding is also not the best choice for predicting the visual quality.

6.1.2. Bilinear 2D-PCR

The second method to perform two-way data analysis on three-way data discussed in this thesis is the bilinear *two-dimensional principal component regression* (2D-PCR). It is based on the two-dimensional principal competent analysis (2D-PCA) proposed by Yang et al. [371] in the context of face recognition and representation. The idea is not to perform the PCA on the temporally pooled features in the two-way array $\mathbf{X}$, but rather to perform the PCA on the average covariance matrix of all frontal slices $\mathbf{X}_t$ of $\underline{\mathbf{X}}$, described with the so-called *scatter* matrix $\mathbf{X}_{Sct} \in \mathbb{R}^{M \times M}$ of $\underline{\mathbf{X}}$ as

$$\mathbf{X}_{Sct} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{X}_t^\top \mathbf{X}_t, \tag{6.1.7}$$

where we used that $\underline{\mathbf{X}}$ is mean centred and therefore the covariance of each slice $\mathbf{X}_t$ is given by $\mathbf{X}_t^\top \mathbf{X}_t$. Therefore $\mathbf{X}_{Sct}$ can be considered as a measurement of the average temporal variation within $\underline{\mathbf{X}}$. Note, that in line with the terminology used in this thesis it should be called rather two-way principal component analysis, as it considers the variation within two different modes by using the scatter matrix. Kong et al. [156] showed that the definition of the scatter matrix $\mathbf{X}_{Sct}$ in (6.1.7) can also be written as

$$\mathbf{X}_{Sct} = \frac{1}{T} \mathbf{X}_{M \times NT} \mathbf{X}_{M \times NT}^\top, \tag{6.1.8}$$

where $\mathbf{X}_{M \times NT} \in \mathbb{R}^{M \times NT}$ represents the mode-2 unfolding of $\underline{\mathbf{X}}$. Hence, the 2D-PCA can be interpreted as essentially the PCA performed on a representation of the unfolded three-way feature array $\underline{\mathbf{X}}$ in the form of the covariance matrix . Compared to the matrix $\mathbf{X}_{N \times MT}$ gained by the mode-1 unfolding of $\underline{\mathbf{X}}$ with $MT \gg M$ columns, the scatter matrix $\mathbf{X}_{Sct}$ has significantly fewer columns, leading to less computational complexity in the determination of the principal components. Besides face recognition and representation as in [156, 371], 2D-PCA has also been used in the audio domain by Rothbucher et al. in [274, 275].

Although 2D-PCA allows us to describe the variation within $\underline{\mathbf{X}}$ better compared to a PCA on the temporally pooled features in $\mathbf{X}$, we are interested in building a prediction model for the visual quality of unknown video sequences. In [148] and [146], I therefore extended the 2D-PCA to the *two-dimensional principal component regression* (2D-PCR). It is based on the standard two-way PCR dicussed in Section 5.4, but takes the three-way structure of the data into account. As in the 2D-PCA, firstly the scatter matrix $\mathbf{X}_{Sct}$ is calculated, followed by a subsequent PCA on $\mathbf{X}_{Sct}$, resulting in the loadings $\mathbf{P} \in \mathbb{R}^{M \times R}$ and the scores $\mathbf{T}_{Sct} \in \mathbb{R}^{M \times R}$ for the scatter matrix. Note, that unlike for the PCA on the two-way feature array $\mathbf{X}$ resulting in scores $\mathbf{t}_r$ of dimension N, the scores $\mathbf{t}_{Sct,r}$ of $\mathbf{T}_{Sct}$ have the dimension M due to the square nature of the covariance matrix $\mathbf{X}_t^\top \mathbf{X}_t$ in (6.1.7).

The idea behind 2D-PCR is to use the explanation of the variance of $\mathbf{X}_{Sct}$ contained in the extracted R principal components and corresponding loadings $\mathbf{P}$ to build a prediction model for each of the frontal slices $\mathbf{X}_t$ of $\underline{\mathbf{X}}$. 2D-PCR is an iterative algorithm for each frontal slice $\mathbf{X}_t$ of $\underline{\mathbf{X}}$ and described in Algorithm 6.2 on the facing page.

Algorithm 6.2: 2D-PCR

Calculate scatter matrix $\mathbf{X}_{Sct}$

1 $\quad \mathbf{X}_{Sct} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{X}_t^\top \mathbf{X}_t$

Perform PCA on $\mathbf{X}_{Sct}$ *to determine loadings* $\mathbf{P}$

2 $\quad \mathbf{P} \leftarrow \mathrm{PCA}(\mathbf{X}_{Sct})$

For each frame t

3 **for** $t = 1 \cdots T$ **do**

Determine scores $\mathbf{T}_t$

4 $\quad \mathbf{T}_t = \mathbf{X}_t \mathbf{P}$

Estimate weights $\widehat{\mathbf{q}}_t$

5 $\quad \widehat{\mathbf{q}}_t = (\mathbf{T}_t^\top \mathbf{T}_t)^{-1} \mathbf{T}_t^\top \mathbf{y}$

Estimate weights $\widehat{\mathbf{b}}_t$

6 $\quad \widehat{\mathbf{b}}_t = \mathbf{P} \widehat{\mathbf{q}}_t$

end

Rearrange all $\widehat{\mathbf{b}}_t$ *into* $\widehat{\mathbf{B}}$

7 $\quad \widehat{\mathbf{B}} = \begin{bmatrix} \widehat{\mathbf{b}}_1 & \widehat{\mathbf{b}}_2 & \cdots & \widehat{\mathbf{b}}_t & \cdots & \widehat{\mathbf{b}}_T \end{bmatrix}$

With the loadings $\mathbf{P}$, we can determine the scores $\mathbf{T}_t \in \mathbb{R}^{N \times R}$ for each frontal slice $\mathbf{X}_t$ as

$$\mathbf{T}_t = \mathbf{X}_t \mathbf{P} \tag{6.1.9}$$

and use these scores to obtain a least squares estimate $\widehat{\mathbf{q}}_t \in \mathbb{R}^{R \times 1}$ for the regression weights $\widehat{\mathbf{q}}_t \in \mathbb{R}^{R \times 1}$ of each slice $\mathbf{X}_t$ as

$$\widehat{\mathbf{q}}_t = (\mathbf{T}_t^\top \mathbf{T}_t)^{-1} \mathbf{T}_t^\top \mathbf{y}, \tag{6.1.10}$$

similar to (5.4.19) for the PCR. In contrast to the PCR, however, the loadings, $\mathbf{P}$ are not the loadings gained from the PCA on $\mathbf{X}_t$, but rather from the PCA on $\mathbf{X}_{Sct}$. Hence, we are not projecting $\mathbf{X}_t$ on a subspace explaining the variance of the features of the t-frame best, but rather on a subspace explaining the average covariance of the complete video sequence with all T frames best, with the scores $\mathbf{T}_t$ representing the projection of $\mathbf{X}_t$ on this new subspace. Thus the weights $\widehat{\mathbf{q}}_t$ do not provide a least square estimation of the regression weights with respect to only the features' variation within the current feature slice $\mathbf{X}_t$, but rather with respect to the average covariance within the complete three-way feature array $\underline{\mathbf{X}}$, therefore providing a better inclusion of the temporal variation within a video sequence. The weights $\widehat{\mathbf{b}}_t \in \mathbb{R}^{M \times 1}$ relating each frontal feature slice to the visual quality $\mathbf{y}$ can then be expressed as

$$\widehat{\mathbf{b}}_t = \mathbf{P} \widehat{\mathbf{q}}_t. \tag{6.1.11}$$

For a unknown video sequence $\underline{\mathbf{S}}_U \notin \mathcal{S}_C$ and corresponding two-way feature array $\mathbf{X}_u \in \mathbb{R}^{M \times T}$, the visual quality prediction $\widehat{y}_{u,t}$ for each frame t represented by its columns $\mathbf{x}_{u,t} \in \mathbb{R}^{M \times 1}$ is then given by

$$\widehat{y}_{u,t} = \mathbf{x}_{u,t}^\top \widehat{\mathbf{b}}_t. \tag{6.1.12}$$

The overall visual quality prediction of $\widehat{y}_u$ can then be gained by averaging over all frames

$$\widehat{y}_u = \frac{1}{T}\sum_{t=1}^{T}\widehat{y}_{u,t}. \tag{6.1.13}$$

Similar to (6.1.2) for the prediction with unfolding, we can rearrange the the weight vectors $\widehat{\mathbf{b}}_t$ of each slice into a two-way array $\widehat{\mathbf{B}} \in \mathbb{R}^{M\times T}$ as

$$\widehat{\mathbf{B}} = \begin{bmatrix}\widehat{\mathbf{b}}_1 & \widehat{\mathbf{b}}_2 & \cdots & \widehat{\mathbf{b}}_t & \cdots & \widehat{\mathbf{b}}_T\end{bmatrix}, \tag{6.1.14}$$

where the t-th column of $\widehat{\mathbf{B}}$ represents the weight vector $\widehat{\mathbf{b}}_t$ for the t-th frame. The prediction for the visual quality $\widehat{y}_u$ in (6.1.13) can then be rewritten with the inner product between $\mathbf{X}_u$ and $\widehat{\mathbf{B}}$ as

$$\widehat{y}_u = \frac{1}{T}\langle \mathbf{X}_u, \widehat{\mathbf{B}}\rangle, \tag{6.1.15}$$

where we assume that $\mathbf{X}_u$ has been preprocessed. The weights $\widehat{\mathbf{B}}$ and offset $\widehat{b}_0$ for predicting the visual quality without prepossessing can be determined in a similar way as for the unfolding and subsequent two-way data analysis in (6.1.5).

In [148] and [146], I have shown that 2D-PCR improves significantly on the prediction performance compared to PCR in combination with either temporal pooling or unfolding. Both the results in [146, 148] and in Chapter 9 of this thesis, however, are only empirical proof of the advantage of 2D-PCR compared to PCR and a thorough theoretical discussion concerning the properties of 2D-PCR, explaining the improvement in prediction performance gained with 2D-PCR, is still missing.

Even tough bilinear 2D-PCR captures the temporal properties for the three-way feature array $\underline{\mathbf{X}}$ better than simple unfolding and subsequent two-way data analysis by using the average covariance of $\underline{\mathbf{X}}$ and avoiding the reshaping of the three-way data in $\underline{\mathbf{X}}$ into a two-way array, we are still fitting only a two-way model to three-way data and are therefore not utilising all modes available in the model building process. Hence, the fitted prediction model is unlikely to provide the best prediction performance.

6.2. Multi-way component models

In the previous section the use of two-way data analysis necessitated the unfolding or slice-wise processing of the three-way feature array $\underline{\mathbf{X}}$ in order to gain a two-way representation $\mathbf{X}$ of $\underline{\mathbf{X}}$, so that the two-way methods could be applied. Obviously, it is preferable to avoid this transformation and directly use the three-way data in the data analysis, particularly as this allows us to gain a multi-way model for the multi-way data.

Multi-way component models allow us to obtain a component description of multi-way data in all modes of the multi-way array i.e. a n-way array can be decomposed into components in all n-modes. For the three-way feature array $\underline{\mathbf{X}}$ we therefore gain a model with

components in all three modes of $\underline{\mathbf{X}}$. Two decomposition models are discussed in this section: the more general *Tucker3* model and the more restricted *PARAFAC* model. Although only the decomposition of three-way arrays is discussed, both models can be extended to *n*-way arrays.

6.2.1. Tucker3

The most general model for three-way arrays is the *Tucker3* model proposed by Tucker in [326] based on his earlier ideas in [324, 325]. The Tucker3 model decomposes the three-way feature array $\underline{\mathbf{X}}$ into a set of components in all three modes of $\underline{\mathbf{X}}$, where the number of components in each mode can be different. $\underline{\mathbf{X}} \in \mathbb{R}^{N \times M \times T}$ is decomposed in a *core array* $\underline{\mathbf{G}} \in \mathbb{R}^{R \times S \times U}$ and component matrices along each mode: $\mathbf{A} \in \mathbb{R}^{N \times R}$ for the first mode, $\mathbf{B} \in \mathbb{R}^{M \times S}$ for the second mode and $\mathbf{C} \in \mathbb{R}^{T \times U}$ for the third mode, where R, S and U denote the number of components in the first, second and third mode, respectively. Usually the number for components in each mode is smaller than the dimension of each mode i.e. $R < N$, $S < M$ and $U < T$. Note, that unlike for the two-way data analysis methods discussed in Chapter 5, where the components are the same in the first and second mode, here the number of components for each mode may be different i.e. $R \neq S \neq U$. With the mode-n product the Tucker3 decomposition of $\underline{\mathbf{X}}$ can be expressed as

$$\underline{\mathbf{X}} = \underline{\mathbf{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} + \underline{\mathbf{E}}, \tag{6.2.1}$$

where $\underline{\mathbf{E}}$ is the residual error not modelled by the component model. The approximation $\widehat{\underline{\mathbf{X}}}$ of $\underline{\mathbf{X}}$ by the Tucker3 model as shown in Fig. 6.1 on the next page is then given by

$$\widehat{\underline{\mathbf{X}}} = \underline{\mathbf{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}. \tag{6.2.2}$$

With $\mathbf{a}_r \in \mathbb{R}^{N \times 1}$, $\mathbf{b}_s \in \mathbb{R}^{M \times 1}$ and $\mathbf{c}_u \in \mathbb{R}^{T \times 1}$ as the r-th, s-th and u-th column of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, representing the r-th, s-th and u-th component in the first, second and third mode, respectively, we can write $\widehat{\underline{\mathbf{X}}}$ also as

$$\widehat{\underline{\mathbf{X}}} = \sum_{r=1}^{R} \sum_{s=1}^{S} \sum_{u=1}^{U} g_{rsu} \circ \mathbf{a}_r \circ \mathbf{b}_s \circ \mathbf{c}_u, \tag{6.2.3}$$

and each element $\widehat{x}_{nmt}$ of $\widehat{\underline{\mathbf{X}}}$ is therefore determined as

$$\widehat{x}_{nmt} = \sum_{r=1}^{R} \sum_{s=1}^{S} \sum_{u=1}^{U} g_{rsu} a_{nr} b_{ms} c_{tu}. \tag{6.2.4}$$

The core array $\underline{\mathbf{G}}$ allows the interaction between all components of all modes, compared to the two-way data analysis methods discussed so far, where only the same components for each of the two modes could interact. Hence each element g_{rsu} of $\underline{\mathbf{G}}$ describes with its

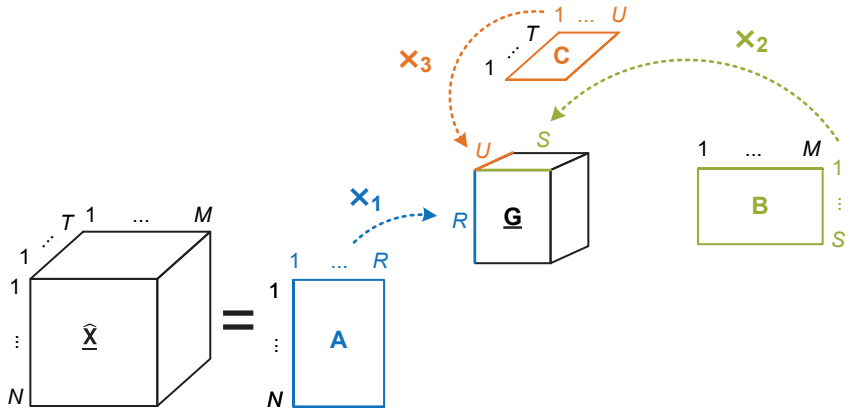

Figure 6.1.: Tucker3 approximation $\underline{\widehat{\mathbf{X}}}$ of the three-way feature array $\underline{\mathbf{X}}$ with core array $\underline{\mathbf{G}}$ and component matrices **A**, **B** and **C**

magnitude the interaction between the r-th component in the first mode, the s-th component in the second mode and the u-th component in the third mode. Using unfolding along the first mode and the Kronecker product, (6.2.2) can be expressed as

$$\widehat{\mathbf{X}}_{N\times MT} = \mathbf{A}\mathbf{G}_{R\times SU}(\mathbf{C}\otimes\mathbf{B})^{\top}, \tag{6.2.5}$$

and the unfolding along the other modes can be expressed similarly [155]. For a multi-way array $\underline{\mathbf{X}}$ an exact Tucker3 decomposition can be determined, if the rank of each component matrix **A**, **B** or **C** corresponds to the *n-rank* of $\underline{\mathbf{X}}$, where the n-rank is defined as the (column-)rank of the mode-n unfolding of $\underline{\mathbf{X}}$ [162]. The Tucker3 decomposition is not unique and the core array $\underline{\mathbf{G}}$ can be modified by appropriate transformations into a desired, possibly more simple structure without affecting the fit of the model as long as the inverse transformations are applied to the component matrices **A**, **B** and **C**. It can also be shown, that the Tucker3 decomposition provides unique subspaces of $\underline{\mathbf{X}}$, regardless of the structure chosen for $\underline{\mathbf{G}}$ in the Tucker3 model [298].

The two most well-known algorithms commonly used to calculate the core array $\underline{\mathbf{G}}$ and the component matrices **A**, **B** and **C** are the algorithms proposed in Tucker's original contribution as *Method I* [326] and the iterative *alternating least squares* (ALS) algorithm *TUCKALS3* proposed by Kroonenberg and de Leeuw [157], described in Algorithm 6.3 on the facing page and Algorithm 6.4 on page 100, respectively. Both algorithms are based on applying the SVD on unfolded representations of the three-way array $\underline{\mathbf{X}}$ along all there modes and provide component matrices **A**, **B** and **C** with orthonormal columns. As each extracted component corresponds to a non-zero singular vector of the unfolded $\underline{\mathbf{X}}$ and there are I_n such non-zero singular vectors in the mode-n unfolded $\underline{\mathbf{X}}$, we have at most I_n components and therefore columns in **A**, **B** or **C**. Note, that I_n can be different for each of the n modes and therefore for a three-way array $I_1 \neq I_2 \neq I_3$. The difference between Tucker's Method I and TUCKALS3 is then that if less than the I_n components in each of the n modes are

Algorithm 6.3: HOSVD for three-way arrays or Tucker's *Method I* [326]

Calculate component matrix $\mathbf{A}$ *of first mode*

1 $\mathbf{A} \leftarrow$ first R left singular vectors of $\mathbf{X}_{N \times MT}$

Calculate component matrix $\mathbf{B}$ *of second mode*

2 $\mathbf{B} \leftarrow$ first S left singular vectors of $\mathbf{X}_{M \times NT}$

Calculate component matrix $\mathbf{C}$ *of third mode*

3 $\mathbf{C} \leftarrow$ first U left singular vectors of $\mathbf{X}_{T \times NM}$

Determine core tensor $\underline{\mathbf{G}}$

4 $\underline{\mathbf{G}} = \underline{\mathbf{X}} \times_1 \mathbf{A}^\top \times_2 \mathbf{B}^\top \times_3 \mathbf{C}^\top$

used for the approximation of $\underline{\mathbf{X}}$, the approximation with Tucker's Method I is not optimal in a least square sense, whereas the approximation with TUCKALS3 is due to its alternating least squares optimisation. In the ALS optimisation each component matrix is successively estimated in a least square sense by assuming the other component matrices to be fixed. This process is repeated until the change of the elements in the core array $\underline{\mathbf{G}}$ and therefore the change in the fit of the model is below a chosen threshold compared to the previous iteration as illustrated in Algorithm 6.4 on the following page. Using less than I_1, I_2 or I_3 components by truncating the components in the different modes after R, S or U components i.e. $R < I_1$, $S < I_2$ or $U < I_3$, the resulting component matrices $\mathbf{A}$, $\mathbf{B}$ or $\mathbf{C}$ have a lower rank compared to the mode-n rank of $\underline{\mathbf{X}}$ with respect to their corresponding modes and we gain a reduced rank approximation $\widehat{\underline{\mathbf{X}}}$ of $\underline{\mathbf{X}}$, similar to the reduced rank approximation for two-way arrays.

Considering the similarities of the Tucker3 decomposition to established two-way methods, the Tucker3 decomposition is also called *higher-order SVD* (HOSVD) [169, 171] or considering the least-squares approximation with TUCKALS3 *three-mode principal component analysis* [157, 159]. In [169], de Lathauwer et al. also showed that Tucker's Method I can in fact be considered as a generalisation of the two-way SVD to three-way arrays. Compared to the SVD for two-way arrays with $\mathbf{X} = \mathbf{USV}^\top$, the core array $\underline{\mathbf{G}}$ can therefore be considered equivalent to $\mathbf{S}$ in representing the singular values and thus the variation within $\underline{\mathbf{X}}$ in terms of new coordinates represented by the component matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ that are similarly corresponding to $\mathbf{U}$ and $\mathbf{V}$ for two-way arrays. Moreover, using the above interpretation of the Tucker3 decomposition as a three-mode principal component analysis, we can also consider $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ as loadings and therefore bases of subspaces that explain the variation of the three-way feature array $\underline{\mathbf{X}}$ in the first, second and third mode best. Other, more efficient algorithms [155] for determining the Tucker3 decomposition of three-way arrays not discussed here include the *higher-order orthogonal iteration* (HOOI) by de Lathauwer et al. [170] and an approach by Eldén and Savas [53] based on Newton-Grassmann optimisation. Improving the speed of ALS-based algorithms in general is discussed in [4, 31].

Complementing the *Tucker3* model are the *Tucker2* and *Tucker1* models. Whereas the

Algorithm 6.4: TUCKALS3 [157]

Initialise component matrices **B***,* **C** with Tucker's Method I

B, **C** $\leftarrow$ first S,U left singular vectors of $\mathbf{X}_{M\times NT}$,$\mathbf{X}_{T\times NM}$

Perform i iterations until change between fits is below threshold θ

repeat

  Calculate component matrix **A** *of first mode*

    A $\leftarrow$ first R left singular vectors of $\mathbf{X}_{N\times MT}(\mathbf{C}\otimes\mathbf{B})$

  Calculate component matrix **B** *of second mode*

    B $\leftarrow$ first S left singular vectors of $\mathbf{X}_{M\times NT}(\mathbf{C}\otimes\mathbf{A})$

  Calculate component matrix **C** *of third mode*

    C $\leftarrow$ first U left singular vectors of $\mathbf{X}_{T\times NM}(\mathbf{B}\otimes\mathbf{A})$

  Determine core tensor $\underline{\mathbf{G}}$

    $\underline{\mathbf{G}} = \underline{\mathbf{X}} \times_1 \mathbf{A}^\top \times_2 \mathbf{B}^\top \times_3 \mathbf{C}^\top$

  Calculate change δ to previous iteration

    $\delta = \sum_{r,s,u} g^2_{rsu,i-1} - g^2_{rsu,i}$

until $\delta < \theta$

Tucker3 decomposition decomposes $\underline{\mathbf{X}}$ into three components, one for each of the three modes in $\underline{\mathbf{X}}$, the Tucker2 and Tucker1 decompositions decompose $\underline{\mathbf{X}}$ into only two and one component, respectively. The modes of $\underline{\mathbf{X}}$ that are not decomposed into components are contained within the core array $\underline{\mathbf{G}}$ and can be chosen freely depending on the desired structure of the resulting component model of $\underline{\mathbf{X}}$. The non-unique nature of Tucker3 models also extends to the Tucker2 and Tucker1 decomposition. Choosing the third mode corresponding to the temporal mode of the feature array $\underline{\mathbf{X}}$, we can express the approximation $\widehat{\underline{\mathbf{X}}}$ of $\underline{\mathbf{X}}$ with the Tucker2 decomposition as

$$\widehat{\underline{\mathbf{X}}} = \underline{\mathbf{G}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{I}_T, \tag{6.2.6}$$

where $\mathbf{I}_T \in \mathbb{R}^{T\times T}$ is the identity matrix replacing the component matrix **C** of the third mode and the core array $\underline{\mathbf{G}}$ is now $\underline{\mathbf{G}} \in \mathbb{R}^{R\times S\times T}$ or using the mode-1 unfolding this can also be expressed as

$$\widehat{\mathbf{X}}_{N\times MT} = \mathbf{A}\mathbf{G}_{R\times ST}(\mathbf{I}\otimes\mathbf{B})^\top. \tag{6.2.7}$$

Note, that the third mode of **X** is no longer reduced and therefore included in the core array $\underline{\mathbf{G}}$ as the third mode. This can be extended straightforwardly to the two other possible Tucker2 decompositions, where either the first or second mode is not reduced. If we exclude both the second and third mode in the Tucker1 decompostion of $\underline{\mathbf{X}}$ corresponding to the feature and temporal mode of the feature array $\underline{\mathbf{X}}$, we can express $\widehat{\underline{\mathbf{X}}}$ as

$$\widehat{\underline{\mathbf{X}}} = \underline{\mathbf{G}} \times_1 \mathbf{A} \times_2 \mathbf{I}_M \times_3 \mathbf{I}_T, \tag{6.2.8}$$

where $\mathbf{I}_T \in \mathbb{R}^{T\times T}$ is the identity matrix replacing the component matrix **C** of the third mode, $\mathbf{I}_M \in \mathbb{R}^{M\times M}$ is the identity matrix replacing the component matrix **B** of the second mode

and the core array $\underline{\mathbf{G}}$ is now $\underline{\mathbf{G}} \in \mathbb{R}^{R \times M \times T}$ or using the mode-1 unfolding this can also be expressed as

$$\begin{aligned} \widehat{\mathbf{X}}_{N \times MT} &= \mathbf{A}\mathbf{G}_{R \times MT}(\mathbf{I}_T \otimes \mathbf{I}_M)^\top \\ &= \mathbf{A}\mathbf{G}_{R \times MT}. \end{aligned} \tag{6.2.9}$$

Hence the second and third mode are no longer reduced and therefore included in the core array $\underline{\mathbf{G}}$. The reduction of other modes for the Tucker1 decomposition can be achieved by replacing the appropriate modes. An element-wise and outer product definition of the Tucker2 and Tucker1 decomposition can be expressed similar to the corresponding definitions for the Tucker3 decomposition in (6.2.3) and (6.2.4). For the Tucker2 decomposition the algorithms for the Tucker3 decomposition can be adapted by replacing the component matrix corresponding to the non-reduced mode with the appropriate identity matrix. The Tucker1 decomposition is equivalent to the PCA on the unfolded representations of $\underline{\mathbf{X}}$ as can easily be seen from Algorithm 6.3 and therefore the common algorithms for two-way arrays as discussed in Section 5.4 can be used.

Comparing the Tucker3, Tucker2 and Tucker1 decompositions, a hierarchy can be established, where a Tucker1 model of a given three-way array can be considered as the least restricted model and the Tucker3 model as the most restricted model. Restriction in this context means that due to the explicit parametrisation of a three-way array in its modes, the more modes are parametrised, the more constrained the resulting approximation is. Hence a model compressing less modes allows for more flexibility and therefore a better fit of the approximation, but at the same time leads to a more complex model. Under the assumption that the same number of components is used and the data has a three-was structure that can be exploited by a three-way model, the increased fit of the less restricted models does not necessarily describe the systematic variation better and may only describe noise, as all three-way models should be capable to describe the systematic variation [298]. Considering that the Tucker1 decomposition is the least restricted Tucker decomposition, but also equivalent to the application of two-way PCA on an unfolded three-way array, this supports at least for the two-way PCA the argument in Section 6.1.1 that unfolding is usually not the best option to handle data with three-way structure.

6.2.2. PARAFAC

Parallel factors (PARAFAC) proposed by Harshman in [82] can be considered as a further restriction of the Tucker3 model. The concept behind PARAFAC was independently also proposed by Carroll and Chang as *canonical decomposition* (CANDECOMP) in [37] and is therefore sometimes referred to as CANDECOMP/PARAFAC or CP decompostion e.g. in [152, 155]. Both are based on earlier concepts by Hitchcock [86, 87] and Cattell [38, 39]. The main difference to the Tucker3 model is that the number of components in each mode is the same and that the approximation $\widehat{\underline{\mathbf{X}}}$ of $\underline{\mathbf{X}}$ is expressed as a sum of *triads* generated with the components, therefore resulting in a *trilinear* model. This is equivalent to the

Tucker3 decomposition with the same number of components in each mode i.e. $R = S = U$ and a corresponding *superdiagonal* core array $\underline{\mathbf{G}} \in \mathbb{R}^{R \times S \times U}$ where all off-superdiagonal elements are zero and all superdiagonal elements are one i.e. $g_{rsu} = 0$ for $r \neq s \neq u$ and $g_{rsu} = 1$ for $r = s = u$. Hence the core array $\underline{\mathbf{G}}$ can be replaced with the identity three-way array $\underline{\mathbf{I}} \in \mathbb{R}^{R \times R \times R}$ and with component or loading matrices along each mode $\mathbf{A} \in \mathbb{R}^{N \times R}$, for the first mode, $\mathbf{B} \in \mathbb{R}^{M \times R}$ for the second mode and $\mathbf{C} \in \mathbb{R}^{T \times R}$ for the third mode as before in the Tucker3 model, we can express the PARAFAC decomposition of $\underline{\mathbf{X}}$ similar to (6.2.1) as

$$\underline{\mathbf{X}} = \underline{\mathbf{I}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} + \underline{\mathbf{E}} \tag{6.2.10}$$

and the approximation $\widehat{\underline{\mathbf{X}}}$ of $\underline{\mathbf{X}}$ similar to (6.2.2) as

$$\widehat{\underline{\mathbf{X}}} = \underline{\mathbf{I}} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}. \tag{6.2.11}$$

Using the superdiagonality of $\underline{\mathbf{I}}$, each element $\widehat{x}_{nmt}$ of $\widehat{\underline{\mathbf{X}}}$ is then given by

$$\widehat{x}_{nmt} = \sum_{r=1}^{R} a_{nr} b_{mr} c_{tr}. \tag{6.2.12}$$

With $\mathbf{a}_r \in \mathbb{R}^{N \times 1}$, $\mathbf{b}_r \in \mathbb{R}^{M \times 1}$ and $\mathbf{c}_r \in \mathbb{R}^{T \times 1}$ as the r-th of column of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, representing the r-th component in the first, second and third mode, respectively, we can write $\widehat{\underline{\mathbf{X}}}$ also as

$$\widehat{\underline{\mathbf{X}}} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \tag{6.2.13}$$

which is the commonly used definition of the PARAFAC decomposition as a sum of triads generated by the outer product of the component vectors. The resulting PARAFAC model is illustrated in Fig. 6.2 on the next page. Using unfolding along the first mode and the Khatri-Rao product, (6.2.13) can be expressed as

$$\widehat{\mathbf{X}}_{N \times MT} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\top, \tag{6.2.14}$$

and the unfolding along the other modes can be expressed similarly [155].

The PARAFAC model is directly linked to the concept of the rank of a three-way array that is defined similar to the rank of matrices as the smallest number of rank-one three-way arrays represented by triads that generate $\underline{\mathbf{X}}$ as their sum [161]. As the PARAFAC model decomposes $\underline{\mathbf{X}}$ into triads, the smallest number of I components in a PARAFAC model that provides an exact fit for $\underline{\mathbf{X}}$ resulting in $\widehat{\underline{\mathbf{X}}} = \underline{\mathbf{X}}$ is therefore equivalent to the rank of $\underline{\mathbf{X}}$ i.e. rank($\underline{\mathbf{X}}$) = I. Thus a PARAFAC model is the best low-rank approximation of a three-way array and if we truncate the approximation of $\underline{\mathbf{X}}$ after R components with $R < I$, the PARAFAC model is then the best rank-R approximation $\widehat{\underline{\mathbf{X}}}$ of $\underline{\mathbf{X}}$. As the components in each mode can not be assumed to be orthogonal, the rank-R approximation of $\underline{\mathbf{X}}$ can not be constructed by adding R triads from separate, iteratively gained rank-1 approximations

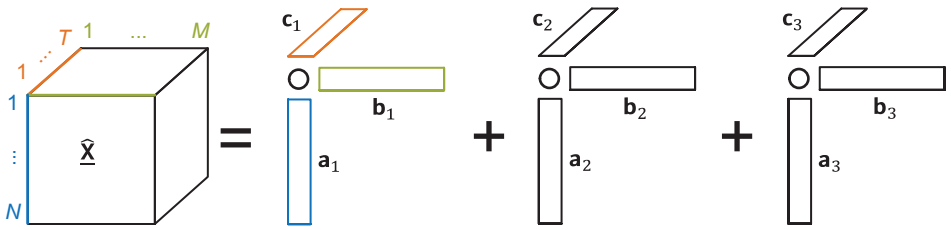

Figure 6.2.: PARAFAC decompostion of the three-way feature array $\underline{\mathbf{X}}$ with $R = 3$ components and corresponding component vectors $\mathbf{a}_r$, $\mathbf{b}_r$ and $\mathbf{c}_r$

corresponding to the r-th component as is for example done in the NIPALS algorithm [180]. Additionally, in some cases the best rank-R approximation can not even be represented by the sum of the R rank-1 three-way arrays generated by the triads from simultaneously extracted components [154] and these *degenerate* solutions usually occur if the three-way arrays is not suitable for the approximation by a trilinear model [298]. In general, however, if no exactly fitting PARAFAC model exists, there are no straightforward algorithms to determine the rank of a three-way array and this problem has been shown to be NP-hard [93]. Therefore usually the rank is determined numerically by iteratively fitting PARAFAC models with an increasing number of components R until the fit of the model is considered to be good enough, but especially for noisy data a perfect fit may not be achievable and therefore the exact rank can not be determined [155]. Although the exact rank of a three-way array may not be determinable, it is sometimes possible to provide upper bounds of the achievable rank, the *maximum rank*, or provide a range of the most probable ranks, the *typical rank*, and I refer to Kolda and Bader [155] for a in-depth discussion of this issue.

Similar to the Tucker3 decomposition, the PARAFAC decompostion can be determined using an alternating least squares approach and the most common algorithm is still the original approach proposed by Harshman in [82] as described in Algorithm 6.5 on the following page, where each component matrix is successively estimated in a least square sense by assuming the other component matrices to be fixed and this is repeated until the change in the fit of the model is below a chosen threshold compared to the previous iteration. Note that all components of a mode need to be determined simultaneously as each mode's components can not be assumed to be orthogonal. Although additional constraints can be introduced into the ALS algorithm resulting in orthonormal loadings, this will result in a reduced model fit and moreover the orthogonality constraint is limited to two of the three modes as the loss of fit in the constraint modes needs to be compensated in the third and unconstrained mode [30]. $\mathbf{B}$ and $\mathbf{C}$ are initialised with the first R singular vectors

Algorithm 6.5: PARAFAC ALS [82]

Initialise component matrices $\mathbf{B}$*,* $\mathbf{C}$

1 $\mathbf{B}, \mathbf{C} \leftarrow$ first R left singular vectors of $\mathbf{X}_{M\times NT}$,$\mathbf{X}_{T\times NM}$

Perform i iterations until change between fits is below threshold θ

2 **repeat**

Calculate component matrix $\mathbf{A}$ *of first mode*

3 $\mathbf{A} = \mathbf{X}_{N\times MT}(\mathbf{C} \odot \mathbf{B})(\mathbf{C}^\top\mathbf{C} \circledast \mathbf{D}^\top\mathbf{D})^+$

Calculate component matrix $\mathbf{B}$ *of second mode*

4 $\mathbf{B} = \mathbf{X}_{M\times NT}(\mathbf{C} \odot \mathbf{A})(\mathbf{C}^\top\mathbf{C} \circledast \mathbf{A}^\top\mathbf{A})^+$

Calculate component matrix $\mathbf{C}$ *of third mode*

5 $\mathbf{C} = \mathbf{X}_{T\times NM}(\mathbf{B} \odot \mathbf{A})(\mathbf{B}^\top\mathbf{B} \circledast \mathbf{A}^\top\mathbf{A})^+$

Calculate change δ *in model fit to previous iteration*

6 $\delta = \|\widehat{\underline{\mathbf{X}}}_{i-1} - \widehat{\underline{\mathbf{X}}}_i\|$

until $\delta < \theta$

of the corresponding unfolding of $\underline{\mathbf{X}}$ as a reasonable starting point for the iterative process, but for some three-way arrays this may not lead to a optimum solution and I refer to Smilde et al. [298] for further elaborations on this issue. Other algorithms for PARAFAC aim to either increase the speed e.g. the alternating slice-wise diagonalisation (ASD) by Jiang et al. [127] that uses slice-wise diagonalisation in combination with a subsequent SVD on the slices of the three-way arrays or to optimise the component matrices not iteratively, but simultaneously e.g. the damped Gauss-Newton method based PMF3 algorithm by Paatero [236], and I refer to the surveys by Faber et al. [56] and Tomasi and Bro [321] for a detailed comparison of different PARAFAC algorithms.

Unlike the Tucker3 decomposition, the PARAFAC decomposition is usually unique upon scaling and permutation [293]. Kruskal [161] has shown that a sufficient criterion for the uniqueness of the trilinear decompostion of a three-way array $\underline{\mathbf{X}}$ is given by rank($\mathbf{A}$) + rank($\mathbf{B}$) + rank($\mathbf{C}$) $\geq 2R + 2$. Additionally, not only the subspaces defined by the loadings $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{A}$ are unique, but also the orientation of the subspaces' basis vectors [298]. One might therefore assume that the PARAFAC decompostion not only provides the best low-rank approximation of a three-way array, but also that the joint subspaces described by the loadings $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ provide the best subspace approximation with respect to the explained variance. It can, however, be shown that the projection of three-way arrays on subspaces represented by the loading matrices of a PARAFAC decomposition equals a Tucker3 model and I refer to the proof by Bro et al. [33] discussed in Appendix A.3.1 for the details. Thus even tough PARAFAC provides the best low-rank approximation of a three-way array, a PARAFAC model does not provide the best subspace approximation of a three-way array with respect to the variation explained by the subspaces and the best subspace approximation is instead provided by a Tucker3 model. This is in contrast to the two-way case, where the PCA represents both the best low-rank and subspace approxima-

tion. Considering that the PARAFAC model is a restricted Tucker3 model, we can extend the hierarchy for Tucker decompositions discussed previously to include PARAFAC, with PARAFAC as the most and Tucker1 as the least restricted decomposition [151]. Similar to Tucker2 and Tucker1 models that provide a better fit than a Tucker3 model, the Tucker3 model will provide a better fit than a PARAFAC model, but at the same time possibly include more noise in its structure [298].

Scores and Loadings For multi-way component models, we have so far treated all modes equally and considered the component matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ as loading matrics of the first, second and third mode, respectively. We can, however, also extend the concept of the two-way component model in Section 5.3 with loadings $\mathbf{P}$ as a projection of the original variables into a new subspace and the scores $\mathbf{T}$ as the representation of the samples in these new subspace defined by the loadings $\mathbf{P}$ to multi-way component models [298]. As for two-way component models, we assume that the first mode represents the sample mode where each sample describes one of the N different sequences in the feature three-way array $\underline{\mathbf{X}}$. Using the Tucker1 decompostion of the mode-1 unfolded three-way array $\underline{\mathbf{X}}$ from (6.2.9) as $\widehat{\mathbf{X}}_{N\times MT} = \mathbf{A}\mathbf{G}_{R\times MT}$ and replacing $\mathbf{A}$ with $\mathbf{T}$ and $\mathbf{G}_{R\times MT}$ with $\mathbf{P}^\top$, we can rewrite (6.2.9) as

$$\widehat{\mathbf{X}}_{N\times MT} = \mathbf{T}\mathbf{P}^\top, \tag{6.2.15}$$

which clearly describes a two-way component model as discussed in Section 5.3. Extending this approach also to the Tucker2, Tucker3 and PARAFAC decomposition, we can write the scores of the three-way component models as

$$\mathbf{T} = \mathbf{A} \tag{6.2.16a}$$

and the loadings as

$$\mathbf{P}^\top_{Tucker1} = \mathbf{G}_{R\times MT} \tag{6.2.16b}$$

$$\mathbf{P}^\top_{Tucker2} = \mathbf{G}_{R\times ST}(\mathbf{I}\otimes\mathbf{B})^\top \tag{6.2.16c}$$

$$\mathbf{P}^\top_{Tucker3} = \mathbf{G}_{R\times SU}(\mathbf{C}\otimes\mathbf{B})^\top \tag{6.2.16d}$$

$$\mathbf{P}^\top_{PARAFAC} = (\mathbf{C}\odot\mathbf{B})^\top \tag{6.2.16e}$$

that is following straightforwardly from (6.2.5), (6.2.7), (6.2.9) and (6.2.14). Hence three-way component models also can be interpreted as a two-way component model where the structure of loadings $\mathbf{P}$ is depending on the chosen multi-way component model [298].

Due to the trilinear structure of the PARAFAC model, we can directly map the component matrices and loadings in the second and third $\mathbf{B}$ and $\mathbf{C}$ onto loadings $\mathbf{P}^M$ and $\mathbf{P}^T$, respectively, where following the suggestion from Smilde et al. [298] the superscript denotes the mode of the loading. We can then express the approximation of a mode-1 unfolded three-way array $\underline{\mathbf{X}}$ with the PARAFAC decomposition as

$$\widehat{\mathbf{X}}_{N\times MT} = \mathbf{T}(\mathbf{P}^T\odot\mathbf{P}^M)^\top. \tag{6.2.17}$$

Note that this direct mapping is not possible for the Tucker3 and Tucker2 models due to the core array $\underline{\mathbf{G}}$ and the interactions between the different modes described within it.

Using multi-way component models to represent multi-way arrays, two-way data analysis methods can then be extended to three-way arrays and for more information on the discussed multi-way component models, I refer to Kolda and Bader [155] and the references therein. Kroonenberg also provides a comprehensive bibliography and list of resources in [158].

6.3. Trilinear PLSR

Multilinear PLS is an extension of the two-way partial least squares concept to multi-way arrays and was first proposed by Bro in [28, 29], with further refinements later proposed by Smilde [299] and de Jong [130]. Reflecting the multi-way nature with n different modes, multilinear PLS is also called *N-PLS* and the notation of the two-way bilinear PLS is extended with a prefix indicating the number of modes in the independent variables. Therefore the univariate PLSR between a three-way feature array $\underline{\mathbf{X}}$ and a visual quality one-way array $\mathbf{y}$ is denoted as tri-PLS1 or, in line with the notation used in this thesis for the bilinear PLS, the *trilinear PLSR*. Clearly, the same arguments in favour of the bilinear PLSR also hold true for the trilinear PLSR, in particular the advantage of the integral consideration of the covariance between $\underline{\mathbf{X}}$ and $\mathbf{y}$ in the model building process. As the trilinear PLSR has been so far the multi-way prediction method studied most in literature, I focus the discussion of multi-way data analysis methods in this thesis on the three-way version of the multilinear PLSR and other multi-way prediction models are only reviewed briefly.

Before discussing the trilinear PLSR, it should be noted, that Wold et al. proposed a *multi-way PLS* earlier in [363], where Wold et al. used a Tucker1 decomposition of the multi-way array, followed by a bilinear PLSR. It thus corresponds to a PLSR on the unfolded representation of a multi-way array. Although a Tucker1 decomposition leads to a valid multi-way component model, its assumptions about and utilisation of the inherent multi-way structure in the data are the weakest of all Tucker models as discussed in Section 6.2. Hence Smilde et al. [298] argue that it should therefore not be considered as multi-way PLS in order to avoid confusion.

General idea Considering the three-way feature array $\underline{\mathbf{X}} \in \mathbb{R}^{N \times M \times T}$ and the subjective visual quality vector $\mathbf{y} \in \mathbb{R}^N$, the aim is to replace the bilinear structure of scores $\mathbf{t}_r \in \mathbb{R}^N$ and weights $\mathbf{w}_r \in \mathbb{R}^M$ for the r-th component in the bilinear PLSR by a trilinear structure of scores $\mathbf{t}_r \in \mathbb{R}^N$, weights $\mathbf{w}_r^M \in \mathbb{R}^M$ and weights $\mathbf{w}_r^M \in \mathbb{R}^T$ representing the different sequences in the first mode, the features in the second mode and frames in the third mode, respectively. The required trilinearity directly implies a three-way one component PARAFAC model for each trilinear PLSR component. For the mode-1 unfolded three-way

feature array $\underline{\mathbf{X}}$ a PARAFAC rank-1 approximation $\widehat{\mathbf{X}}_{N\times MT}$ can be expressed as

$$\widehat{\mathbf{X}}_{N\times MT,r} = \mathbf{t}_r(\mathbf{w}_r^T \odot \mathbf{w}_r^M)^\top = \mathbf{t}_r(\mathbf{w}_r^T \otimes \mathbf{w}_r^M)^\top, \tag{6.3.1}$$

where we used that for vectors the Khatri-Rao product is corresponding to the Kronecker product. It can be shown [299] that the scores $\mathbf{t}_r$ given by

$$\mathbf{t}_r = \mathbf{X}_{N\times MT,r}(\mathbf{w}_r^T \otimes \mathbf{w}_r^M), \tag{6.3.2}$$

with weights $\mathbf{w}_r^M$ and $\mathbf{w}_r^T$ of length one i.e. $\|\mathbf{w}^M\|, \|\mathbf{w}^T\| = 1$ solve the minimisation problem

$$\min_{\mathbf{t}_r} \|\mathbf{X}_{N\times MT,r} - \mathbf{t}_\mathbf{r}(\mathbf{w}_R^T \otimes \mathbf{w}_r^M)^\top\|^2. \tag{6.3.3}$$

Similar as to the bilinear PLSR, the aim is then to find weights $\mathbf{w}_r^M$ and weights $\mathbf{w}_r^T$ that maximise the covariance between the visual quality vector $\mathbf{y}$ and the scores $\mathbf{t}_r$. Omitting the index r denoting the r-th component for clarity, this can be written for the trilinear PLS as

$$\max_{\mathbf{w}^M,\mathbf{w}^T} \left[\text{cov}(\mathbf{t},\mathbf{y})|\mathbf{t} = \mathbf{X}_{N\times MT}(\mathbf{w}^T \otimes \mathbf{w}^M)\right], \tag{6.3.4}$$

where $\mathbf{w}^M$ and $\mathbf{w}^T$ are again restricted to length one. With $\mathbf{w} = \mathbf{w}^T \otimes \mathbf{w}^M$, the score t_n of the n-th horizontal slice of $\underline{\mathbf{X}}$ as $\mathbf{X}_n \in \mathbb{R}^{M\times T}$, representing the n-th sequence can be expressed as [299]

$$t_n = \mathbf{X}_n\mathbf{w} \tag{6.3.5a}$$

and this equivalent to

$$t_n = (\mathbf{w}^M)^\top\mathbf{X}_n\mathbf{w}^T. \tag{6.3.5b}$$

Using this representation of the scores, the optimisation problem for the covariance between visual quality $\mathbf{y}$ and scores $\mathbf{t}$ in (6.3.4) can then be rewritten as [28]

$$\begin{aligned}
&\max_{\mathbf{w}^M,\mathbf{w}^T}\left[\text{cov}(\mathbf{t},\mathbf{y})|t_n = (\mathbf{w}^M)^\top\mathbf{X}_n\mathbf{w}^T\right]\\
&= \max_{\mathbf{w}^M,\mathbf{w}^T}\left[\sum_{n=1}^N t_n y_n|t_n = (\mathbf{w}^M)^\top\mathbf{X}_n\mathbf{w}^T\right]\\
&= \max_{\mathbf{w}^M,\mathbf{w}^T}\left[\sum_{n=1}^N y_n(\mathbf{w}^M)^\top\mathbf{X}_n\mathbf{w}^T\right]\\
&= \max_{\mathbf{w}^M,\mathbf{w}^T}\left[(\mathbf{w}^M)^\top\left(\sum_{n=1}^N y_n\mathbf{X}_n\right)\mathbf{w}^T\right]\\
&= \max_{\mathbf{w}^M,\mathbf{w}^T}\left[(\mathbf{w}^M)^\top\mathbf{Z}\mathbf{w}^T\right],
\end{aligned} \tag{6.3.6}$$

Algorithm 6.6: Trilinear N-PLS algorithm [28]

$\widetilde{\underline{\mathbf{X}}}_0 = \underline{\mathbf{X}}$
$\widetilde{\mathbf{y}}_0 = \mathbf{y}$
for $r = 1 \ldots R$ **do**
  Calculate $\mathbf{Z}_r$
    $\mathbf{Z}_r = \sum_{n=1}^{N} y_{n,r-1} \widetilde{\mathbf{X}}_{n,r-1}$
  Find weights $\mathbf{w}_r^M$ *and* $\mathbf{w}_r^T$ *by applying the SVD to* $\mathbf{Z}$
    $\mathbf{w}_r^M \leftarrow$ first left singular vector of $\mathbf{Z}_r$
    $\mathbf{w}_r^T \leftarrow$ first right singular vector of $\mathbf{Z}_r$
  Estimate scores $\mathbf{t}_r$
    $\mathbf{t}_r = \widetilde{\mathbf{X}}_{N \times MT, r-1}(\mathbf{w}_r^T \otimes \mathbf{w}_r^M)$
  Estimate regression weights $\widehat{\mathbf{q}}$
    $\widehat{\mathbf{q}}_r = (\mathbf{T}_r^\top \mathbf{T}_r)^{-1} \mathbf{T}_r^\top \widetilde{\mathbf{y}}_{r-1}$
  Subtract the influence of the r-th component from $\widetilde{\underline{\mathbf{X}}}_{r-1}$ *and* $\widetilde{\mathbf{y}}_{r-1}$
    $\widetilde{\mathbf{X}}_{N \times MT, r} = \widetilde{\mathbf{X}}_{N \times MT, r-1} - \mathbf{t}_r(\mathbf{w}_r^T \otimes \mathbf{w}_r^M)$
    $\widetilde{\mathbf{y}}_r = \widetilde{\mathbf{y}}_{r-1} - \mathbf{T}_r \widehat{\mathbf{q}}_r$
end

where $\mathbf{Z} \in \mathbb{R}^{M \times T}$ is given by

$$\mathbf{Z} = \sum_{n=1}^{N} y_n \mathbf{X}_n, \tag{6.3.7}$$

representing the inner product between $\mathbf{y}$ and $\underline{\mathbf{X}}$ i.e. for each element z_{jk} of $\mathbf{Z}$, $z_{jk} = \sum_{n=1}^{N} y_n x_{nmt}$. The solution for this optimisation problem can then be found rather *elegantly* [299] by performing the SVD on $\mathbf{Z}$, resulting in $\mathbf{w}^M$ as the first left singular vector of $\mathbf{Z}$ and $\mathbf{w}^M$ as the first right singular vector of $\mathbf{Z}$.

N-PLS algorithm The trilinear PLSR can be determined with the N-PLS algorithm that was proposed by Bro [28] and is described in Algorithm 6.6. It is based on the non-orthogonal NIPALS PLS1 algorithm by Martens and Næs [196] discussed in Section 5.5 in Algorithm 5.3. The non-orthogonal NIPALS PLS1 algorithm is modified to the now trilinear structure of the scores $\mathbf{t}$ and weights $\mathbf{w}^M$ and $\mathbf{w}^T$, but maintains its iterative structure in finding the R components. The main difference is on the one hand the different estimation of the weights $\mathbf{w}_r^M$ and $\mathbf{w}_r^T$ with the SVD on $\mathbf{Z}_r$ and the trilinear estimation of the scores $\mathbf{t}_r$, and on the other hand that a three-way feature array $\underline{\mathbf{X}}$ is deflated in each iteration r, resulting in the residual three-way array $\widetilde{\underline{\mathbf{X}}}_r$ after the r-th iteration.

Similar to the non-orthogonal NIPALS PLS1 algorithm, the scores $\mathbf{t}_r$ are no longer orthogonal and therefore a simultaneous regression on $\mathbf{y}$ of all r scores determined so far is necessary in each iteration. Hence we estimate in the r-th iteration the regression weights

$\widehat{\mathbf{q}}_r \in \mathbb{R}^{r\times 1}$ in the least square sense with $\mathbf{T}_r \in \mathbb{R}^{N\times r}$, where the columns $\mathbf{t}_1 \ldots \mathbf{t}_r$ represent the scores obtained in all iterations including the current r-th iteration. After R components have been determined, we denote the final scores and regression weights as $\mathbf{T} \in \mathbb{R}^{N\times R}$ and $\widehat{\mathbf{q}} \in \mathbb{R}^{R\times 1}$. The approximation $\widehat{\underline{\mathbf{X}}}_r$ of $\underline{\mathbf{X}}$ given in (6.3.9) achieved after the r-th iteration with N-PLS can also be described with the Khatri-Rao product as

$$\widehat{\mathbf{X}}_{N\times MT,r} = \mathbf{T}_r(\mathbf{W}_r^T \odot \mathbf{W}_r^M)^\top, \tag{6.3.8}$$

where the matrices $\mathbf{W}_r^M \in \mathbb{R}^{M\times r}$ and $\mathbf{W}_r^T \in \mathbb{R}^{T\times r}$ contain in their columns the weights $\mathbf{w}_1^M, \ldots, \mathbf{w}_r^M$ and $\mathbf{w}_1^T, \ldots, \mathbf{w}_r^T$, respectively.

Additionally, we can also define a weight matrix $\mathbf{W} \in \mathbb{R}^{MT\times R}$ with columns $\mathbf{w}_1, \ldots, \mathbf{w}_R$ representing the weights $\mathbf{w}_r$, where $\mathbf{w}_r = \mathbf{w}_r^T \otimes \mathbf{w}_r^M$. Yet, $\mathbf{W}$ is no longer orthogonal as for bilinear PLSR and one consequence is that $\mathbf{T}\mathbf{W}^\top$ in the N-PLS does not provide the least squares fit for (unfolded) $\underline{\mathbf{X}}$ for a given $\mathbf{W}$ as in the non-orthogonal NIPALS PLS1 algorithm. Also even if we can still express the R-component approximation $\widehat{\underline{\mathbf{X}}}$ of $\underline{\mathbf{X}}$ as

$$\widehat{\mathbf{X}}_{N\times MT} = \mathbf{T}\mathbf{W}^\top, \tag{6.3.9}$$

for the scores $\mathbf{T}$ corresponding to $\underline{\mathbf{X}}$ it holds that

$$\mathbf{T} \neq \mathbf{X}_{N\times MT}\mathbf{W}, \tag{6.3.10}$$

due to the non-orthogonality of $\mathbf{W}$. The scores $\mathbf{T}$ can, however, be determined iteratively as is done in the N-PLS algorithm. This can be expressed as a matrix $\mathbf{V} \in \mathbb{R}^{MT\times R}$ [299]

$$\mathbf{V} = \left[\mathbf{w}_1 \quad (\mathbf{I} - \mathbf{w}_1\mathbf{w}_1^\top)\mathbf{w}_2 \quad \cdots \quad \prod_{r=1}^{R-1}(\mathbf{I} - \mathbf{w}_r\mathbf{w}_r^\top)\mathbf{w}_R\right], \tag{6.3.11}$$

and with $\mathbf{V}$ we can determine the scores $\mathbf{T}$ as

$$\mathbf{T} = \mathbf{X}_{N\times MT}\mathbf{V}. \tag{6.3.12}$$

With the scores $\mathbf{T}$, the prediction $\widehat{\mathbf{y}}$ of the visual quality $\mathbf{y}$ is then given by

$$\widehat{\mathbf{y}} = \mathbf{T}\widehat{\mathbf{q}} = \mathbf{X}_{N\times MT}\mathbf{V}\widehat{\mathbf{q}}. \tag{6.3.13}$$

Using (6.3.13), the weights $\widehat{\mathbf{b}} \in \mathbb{R}^{MT}$ relating the mode-1 unfolded three-way feature array $\underline{\mathbf{X}}$ directly to the visual quality vector $\mathbf{y}$ are then given by

$$\widehat{\mathbf{b}} = \mathbf{V}\widehat{\mathbf{q}}. \tag{6.3.14}$$

For the prediction of the visual quality $\widehat{y}_u$ of a unknown video sequence $\underline{\mathbf{S}}_U \notin \mathcal{S}_C$, the mode-1 unfolding is performed on the corresponding horizontal feature slice $\mathbf{X}_u \in \mathbb{R}^{M\times T}$, resulting in a feature vector $\mathbf{x}_u \in \mathbb{R}^{MT}$. The visual quality can then be predicted using $\mathbf{x}_u$ and $\widehat{\mathbf{b}}$ with

$$\widehat{y}_u = \mathbf{x}_u^\top\widehat{\mathbf{b}}, \tag{6.3.15}$$

where we assume that both $\underline{\mathbf{X}}$ and $\mathbf{X}_u$ have been preprocessed. The prediction weights $\widehat{\mathbf{b}}$ for the quality prediction without preprocessing can be determined as described in Section 6.1.2 for the 2D-PCR. Alternatively, it is possible to rearrange the weight vector $\widehat{\mathbf{b}}$ for the unfolded data into a two-way weight array $\widehat{\mathbf{B}} \in \mathbb{R}^{M \times T}$ as described in (6.1.2). The prediction for the visual quality $\widehat{y}_u$ can then be written as the inner product between $\mathbf{X}_u$ and $\widehat{\mathbf{B}}$ as

$$\widehat{y}_u = \langle \mathbf{X}_u, \widehat{\mathbf{B}} \rangle. \tag{6.3.16}$$

Unlike as for the bilinear PLSR, where multiple algorithms with very different properties exist, for the multilinear PLSR in general and the trilinear PLSR in particular so far only the original N-PLS algorithm and its derivations exist. For multi-way arrays with more than three modes, the N-PLS algorithm shown here for three-way arrays can be extended straightforwardly to accommodate the additional modes. Assuming we have a four-way array $\underline{\mathbf{A}} \in \mathbb{R}^{N \times M \times T \times V}$, we can calculate a three-way array $\underline{\mathbf{Z}} \in \mathbb{R}^{M \times T \times V}$ similar to the matrix $\mathbf{Z}$ above, where each element is defined as $z_{mtv} = \sum_{n=1}^{N} y_n a_{nmtv}$. The weights $\mathbf{w}^M$, $\mathbf{w}^T$ and $\mathbf{w}^V$ for each of the three modes can then be determined with a one component trilinear PARAFAC decomposition, enabling the estimation of the scores and I refer to [28, 29] for more details on the multilinear PLSR for n-way arrays. The multilinear PLSR for n-way arrays, however, has so far not received as much attention in literature as the special case of the trilinear PLSR for three-way arrays.

Related to the mulitlinear PLSR and the N-PLS alogrithm is the linear three-way decompostion (LTD) approach by Ståhle [309]. It applies an alternative least square approach for determining the weights $\mathbf{w}^M$ and $\mathbf{w}^T$ by cycling through all modes, representing a hybrid form between Tucker1-PLS as in Wold et al. [363] and the N-PLS [298]. It was shown by de Jong [131] that LTD provided the same weights $\mathbf{w}^M$ and $\mathbf{w}^T$ as the N-PLS algorithm and therefore results in the same prediction results. According to de Jong [131], the N-PLS algorithm has, however, due to its explicit three-way optimisation criterion the advantages of numerically more reliable results and is usually faster compared to the LTD.

Relationship to PARAFAC Even tough the trilinear PLSR uses a three-way PARAFAC model for each of the R components as can be seen in (6.3.1), it is not equivalent to a PARAFAC decomposition of $\underline{\mathbf{X}}$ with R components. The reason for this is that the trilinear PLSR iteratively determines rank-1 approximations for each of the R components and thus the rank-R approximation of $\underline{\mathbf{X}}$ with trilinear PLSR consists of the sum of R separate rank-1 trilinear approximations of $\underline{\mathbf{X}}$ with scores $\mathbf{t}_r$ and weights $\mathbf{w}_r^T$ and $\mathbf{w}_r^M$ for each component. But the PARAFAC decomposition requires the simultaneous estimation of all components and thus the trilinear PLSR is not a PARAFAC decomposition. Hence the rank-R approximation with the the trilinear PLSR components is neither the best rank-R approximation, nor does it provide a unique decomposition of $\underline{\mathbf{X}}$ as both properties require a rank-R PARAFAC model.

Algorithm 6.7: Trilinear N-PLS algorithm without deflation of $\underline{\mathbf{X}}$ [131]

$\widetilde{\mathbf{y}}_0 = \mathbf{y}$
for $r = 1 \ldots R$ **do**
  Calculate $\mathbf{Z}_r$
  $\mathbf{Z}_r = \sum_{n=1}^{N} y_{n,r-1}\mathbf{X}_n$
  Find weights $\mathbf{w}_r^M$ *and* $\mathbf{w}_r^T$ *by applying the SVD to* $\mathbf{Z}$
  $\mathbf{w}_r^M \leftarrow$ first left singular vector of $\mathbf{Z}_r$
  $\mathbf{w}_r^T \leftarrow$ first right singular vector of $\mathbf{Z}_r$
  Estimate scores $\mathbf{t}_r$
  $\mathbf{t}_r = \mathbf{X}_{N\times MT}(\mathbf{w}_r^T \otimes \mathbf{w}_r^M)$
  Estimate regression weights $\widehat{\mathbf{q}}$
  $\widehat{\mathbf{q}}_r = (\mathbf{T}_r^\top \mathbf{T}_r)^{-1}\mathbf{T}_r^\top \widetilde{\mathbf{y}}_{r-1}$
  Subtract the influence of the r-th component from $\widetilde{\mathbf{y}}_{r-1}$
  $\widetilde{\mathbf{y}}_r = \widetilde{\mathbf{y}}_{r-1} - \mathbf{T}_r\widehat{\mathbf{q}}_r$
end

Subspace approximation One consequence of using a PARAFAC decomposition for each component is that the resulting trilinear model for the r-th component does not represent the best subspace approximation of $\underline{\mathbf{X}}$. Therefore Bro et al. [33] proposed a modification of the N-PLS algorithm that extends the multilinear PLSR to provide a subspace approximation of $\underline{\mathbf{X}}$. Bro et al. [33] use that de Jong has shown in [131] that the deflation of $\underline{\mathbf{X}}$ in each iteration of the N-PLS algorithm is not necessary. Hence the model for approximating $\underline{\mathbf{X}}$ with $\widehat{\underline{\mathbf{X}}}$ can be chosen independently from the model imposed on the weights $\mathbf{w}_r^T$ and $\mathbf{w}_r^M$, as only $\underline{\mathbf{X}}$ is used in the estimation of the prediction weight $\widehat{\mathbf{q}}$, but not any intermediate, deflated versions $\widetilde{\underline{\mathbf{X}}}_r$. The resulting N-PLS algorithm without deflation of $\widehat{\underline{\mathbf{X}}}$ is described in Algorithm 6.7. In order to gain a subspace approximation of $\underline{\mathbf{X}}$, the trilinear PARAFAC-like model in (6.3.8) is replaced by a Tucker3 model

$$\widehat{\mathbf{X}}_{N\times MT,r} = \mathbf{T}_r\mathbf{G}_{r\times rr}\left(\mathbf{W}_r^T \otimes \mathbf{W}_r^M\right)^\top, \tag{6.3.17}$$

where the core array $\underline{\mathbf{G}} \in \mathbb{R}^{r\times r\times r}$ is given by

$$\mathbf{G}_{r\times rr} = \mathbf{T}_r^\dagger \mathbf{X}_{N\times MT}\left(\left(\mathbf{W}_r^T\right)^+ \otimes \left(\mathbf{W}_r^M\right)^+\right)^\top. \tag{6.3.18}$$

The advantages of this new Tucker3-based approximation are a better fit of the model i.e. the residual modelling error $\underline{\mathbf{E}} = \underline{\mathbf{X}} - \widehat{\underline{\mathbf{X}}}$ is lower than for the original N-PLS algorithm and that a perfectly trilinear $\underline{\mathbf{X}}$ can be modelled perfectly. At the same time the prediction weights $\widehat{\mathbf{q}}$ remain unchanged and thus this modified version of the N-PLS algorithm provides the same visual quality predictions $\widehat{y}_u$ for unknown video sequences $\underline{\mathbf{S}}_U$ as the original N-PLS algorithm [33].

Alternatives to trilinear PLSR Even tough only the multilinear PLS is discussed in detail in this thesis, other methods for building multi-way models are briefly reviewed and I refer to the given references for further details. As before, the focus is on three-way arrays, but all methods can be extended to n-way arrays.

Using *MLR on a multi-way component model* of the three-way feature array $\underline{\mathbf{X}}$, the visual quality vector $\mathbf{y}$ can be regressed on the mode-1 loading matrix $\mathbf{A}$ representing the scores $\mathbf{T}$ of $\underline{\mathbf{X}}$, similar to the extension from the PCA to the PCR in Section 5.4. For orthogonal scores $\mathbf{T}$, this can be expressed straightforwardly as $\widehat{\mathbf{q}} = \mathbf{T}^\top \mathbf{y}$ and for a Tucker3 model we arrive with (6.2.5) at $\widehat{\mathbf{b}} = \left((\mathbf{C}^+ \otimes \mathbf{B}^+)^\top \mathbf{G}^+_{R\times SU}\right)\widehat{\mathbf{q}}$, allowing us to predict the visual quality of unknown video sequences. But using only the scores gained from a multi-way component model of $\underline{\mathbf{X}}$, the resulting regression weights $\widehat{\mathbf{b}}$ may not necessarily be predictive for $\mathbf{y}$. Hence this approach shares the same disadvantage as the PCR for two-way arrays.

The *multi-way covariate regression* proposed by Smilde and Kiers [300] provides a framework for component model independent regression. Hence the desired model structure can be chosen as one of the three Tucker models or a PARAFAC model, depending on which model may suit the three-way feature array $\underline{\mathbf{X}}$ best. The multi-way covariate regression is based on the principal covariates regression by de Jong and Kiers [133] that allows to adjust the influence of the independent variables represented by $\underline{\mathbf{X}}$ and the dependent variables represented by $\mathbf{y}$ on the sought weights with a parameter α, where $\alpha = 1$, $\alpha = 0.5$ and $\alpha = 0$ can be interpreted as PCR, PLS and MLR, respectively, as can be seen in (6.3.19) below. For a three-way array $\underline{\mathbf{X}} \in \mathbb{R}^{N\times MT}$ and a univariate $\mathbf{y} \in \mathbb{R}^N$, the optimisation problem in the multi-way covariate regression can be expressed as

$$\min_{\mathbf{W}} \left[\alpha \|\mathbf{X}_{N\times MT} - \mathbf{X}_{N\times MT}\mathbf{P}_X^\top\|^2 + \left[(1-\alpha)\|\mathbf{y} - \mathbf{X}_{N\times MT}\mathbf{W}\mathbf{p}_y^\top\|^2\right], \tag{6.3.19}$$

where $\mathbf{W}$ are the weights optimising the expression, $\mathbf{P}_X$ represents the loadings of $\underline{\mathbf{X}}$ for the different multi-way component models from (6.2.16), $\mathbf{p}_y$ the loadings of $\mathbf{y}$ and the scores $\mathbf{T}$ are given by $\mathbf{T} = \mathbf{X}_{N\times MT}\mathbf{W}$. Note, that unlike for the multilinear PLSR in (6.3.4) the structural model of $\underline{\mathbf{X}}$ can be chosen independently from the optimisation problem and can be solved with an alternating least square based algorithm as described in [300]

Recently, Zhao et al. [375] proposed the *higher-order PLS (HOPLS)*, where the concept of covariance has been extended to multi-way arrays with a subsequent Tucker3 decomposition of the resulting covariance multi-way array $\underline{\mathbf{C}}_r$ in order to determine scores $\mathbf{t}_r$ and loadings $\mathbf{P}_r^M$ and $\mathbf{P}_r^T$ for each component r. Unlike the methods discussed so far, the dimensionality of the second mode of the loadings $\mathbf{P}_r^M$ and $\mathbf{P}_r^T$ representing the n-rank is not one, but rather an a-priori design parameter. Hence, for each component r we obtain loadings with an n-rank > 1 or in other words for each component not a loading or weight vector as used so far, but rather a loading matrix is obtained. The first step in the HOPLS algorithm for a three-way $\underline{\mathbf{X}}$ is the optimisation problem

$$\max_{\mathbf{P}_r^M, \mathbf{P}_R^T} \|\underline{\mathbf{C}}_r \times_1 \mathbf{P}_r^M \times_2 \mathbf{P}_r^T\|^2, \tag{6.3.20}$$

that is similar to the maximization of the covariance for the multilinear PLS in (6.3.4). This problem is solved by a Tucker3 decomposition of $\underline{\mathbf{C}}_r$ resulting in core array $\underline{\mathbf{G}}_r$ and the loadings $\mathbf{P}_r^M$ and $\mathbf{P}_r^T$. The scores $\mathbf{t}_r$ are then determined by minimising the following expression

$$\min_{\mathbf{t}_r} \|\underline{\mathbf{X}} - \underline{\mathbf{G}}_r \times_1 \mathbf{t}_r \times_2 \mathbf{P}_r^M \times_3 \mathbf{P}_r^T\|^2, \tag{6.3.21}$$

that can again be compared to the optimisation criteria for the scores of the multilinear PLS in (6.3.3) and where the solution is the first left singular vector of $\underline{\mathbf{X}} \times_2 \left(\mathbf{P}_r^M\right)^\top \times_3 \left(\mathbf{P}_r^T\right)^\top$. Using the scores and loadings, $\underline{\mathbf{X}}$ is than deflated to $\widetilde{\underline{\mathbf{X}}}_r$ and the process is repeated for the $(r+1)$-th component. Considering the similarities to the multilinear PLSR, HOPLS can be understood as a generalization of the multilinear PLSR and it was shown by Zhao et al. in [375] that the multilinear PLSR can be considered as a special case of the HOPLS where the n-rank for the loadings is one. In [375], it is also shown that HOPLS outperforms multilinear PLSR for some datasets and performance metrics, but more comprehensive evaluations especially with commonly used datasets in chemometrics may be necessary before this supposed advantage can be proven to be valid in general.

7. Model Building Considerations

Regardless of which method presented in the previous chapters is chosen to build a prediction model, the training and structure of the model should lead to reliable and well performing prediction models even for unknown video sequences not included in the training.

In this chapter, I therefore address model building considerations that should be taken into account when using the data analysis approach in the design of video quality metrics. Firstly, I discuss the training of reliable prediction models using cross validation, before reviewing the model selection process itself. This is followed by a discussion of influencing the structure of the model and the resulting predication performance by selectively choosing the number of components or specific features.

7.1. Cross validation

Calibration or training results in a model described by the regression weights $\widehat{\mathbf{B}}$, allowing us to estimate the visual quality $\widehat{y}$ purely from the three-way feature array $\underline{\mathbf{X}}$ of a video sequence $\underline{\mathbf{S}}$. But we still need to ensure that the gained model is reliable and therefore able to provide a visual quality prediction $\widehat{y}$ as close as possible to the true subjective visual quality y with respect to a given error metric, often represented by the *mean squared error* between $\widehat{y}$ and y. Hence, the calibration is followed by a *validation*, assessing the prediction error of our model.

Our ultimate goal is the prediction of the visual quality of unknown video sequences and thus the validation should use a different set $\mathcal{S}_V$ of video sequences disjunct to the calibration set $\mathcal{S}_C$ i.e. $\mathcal{S}_C \cap \mathcal{S}_V = \varnothing$ in order to allow the realistic assessment of a model's predictive abilities. The best option is clearly to use a set $\mathcal{S}_P$ of video sequences that is completely separated from the sequences in $\mathcal{S}$ available during the calibration process and thus perform an *external validation* [197]. We can then use all N sequences in $\mathcal{S}$ for the calibration with $\mathcal{S}_C = \mathcal{S}$, followed by using all sequences in $\mathcal{S}_V = \mathcal{S}_P$ for assessing the prediction error of our model. Unfortunately, in video quality assessment the number of available video sequences $\underline{\mathbf{S}}$ and corresponding subjective visual quality values y is usually rather limited with typically $N \leq 50$. Therefore often only one set $\mathcal{S}$ of N video sequences $\underline{\mathbf{S}}_n$ is available both for calibration and validation, forcing us to perform an *internal validation* [197].

Cross validation is a concept that allows us to perform an internal validation by using only video sequences from $\mathcal{S}$. The aim is to split $\mathcal{S}$ into two subsets, one set $\mathcal{S}_C \subset \mathcal{S}$ for calibration and one set $\mathcal{S}_V \subset \mathcal{S}$ for validation, where $\mathcal{S}_C \cup \mathcal{S}_V = \mathcal{S}$, but $\mathcal{S}_C \cap \mathcal{S}_V = \varnothing$

and thus are still disjunct. Depending on the specific cross validation method, different strategies for splitting up $\mathcal{S}$ are used, resulting in different properties of the cross validation methods.

Simple cross validation *Hold-out cross validation* or simple cross validation is the most straightforward method for splitting up $\mathcal{S}$ and can be traced back as early as Larson [168] according to Arlot and Celisse [10]. It splits the set of N sequences into two fixed sets with $|\mathcal{S}_C| = N_c$ calibration and $|\mathcal{S}_V| = N_V$ validation sequences, where $N = N_C + N_V$. Fixed in this context means that once the two sets have been defined, only one model with $\mathcal{S}_C$ will be built, followed by the validation with $\mathcal{S}_V$. The problem with this method is that for a fixed number of video sequences N, a trade off between the number of sequences in the calibration and validation set needs to be made: the larger N_C, the more representative $\mathcal{S}_C$ is of the general population and thus the better the systematic variation can be modelled, but on the other hand only relatively few, $N_V = N - N_C$, sequences are available for the validation and thus $\mathcal{S}_V$ is less representative of the general population, not allowing us to assess the realistic prediction abilities of the model. If we consider cross validation as an estimator for the true, but unknown prediction error for the general population, it can be shown that the bias of the cross validation estimator decreases with increasing N_C, hence motivating $N_C > N_V$, but also that hold-out cross validation in general regardless of the ratio between N_C and N_V results in a relatively large variance compared to other cross validation strategies [10]. Considering these two issues, hold-out cross validation is clearly not the best method and this was also shown empirically for small data sets with $N \leq 80$ samples in chemometrics by Martens and Dardenne [199].

Leave-one-out cross validation *Leave-one-out cross validation* is a cross validation method independently introduced by Allen [2], Geisser [72] and Stone [311] that applies an exhaustive splitting strategy by using $N_C = N - 1$ sequences for calibration and only one sequence for validation with $N_V = 1$. In contrast to the hold-out method, however, not one fixed model is used, but rather N models are constructed, successively leaving out each sequence and using all the other $N - 1$ sequences for calibrating the model as illustrated in Fig. 7.1 on the next page. Thus we have both N calibration sets $|\mathcal{S}_{C,n}| = N-1$ and validation sets $|\mathcal{S}_{V,n}| = 1$, where $\mathcal{S}_{V,n} = \{\underline{\mathbf{S}}_n\}$ and $\mathcal{S}_{C,n} = \mathcal{S} \setminus \{\underline{\mathbf{S}}_n\}$ for $n = 1, \ldots, N$. Considering cross validation again from an estimator point of view, we are now able to use $N_C \approx N$ sequences in the calibration thus reducing the bias of the estimation significantly. Moreover, it can be shown that for leave-one-out cross validation the variance is minimal compared to all other all cross validation strategies [10]. Martens and Dardenne [199] have shown that for the same data set, leave-one-out cross validation provides significantly more realistic results for small data sets compared to the hold-out cross validation. One disadvantage of the leave-one-out cross validation is, however, that now N different models need to be calibrated. But in many cases and also in the context of this thesis, although an increased computational complexity exists, its practical implications are negligible, es-

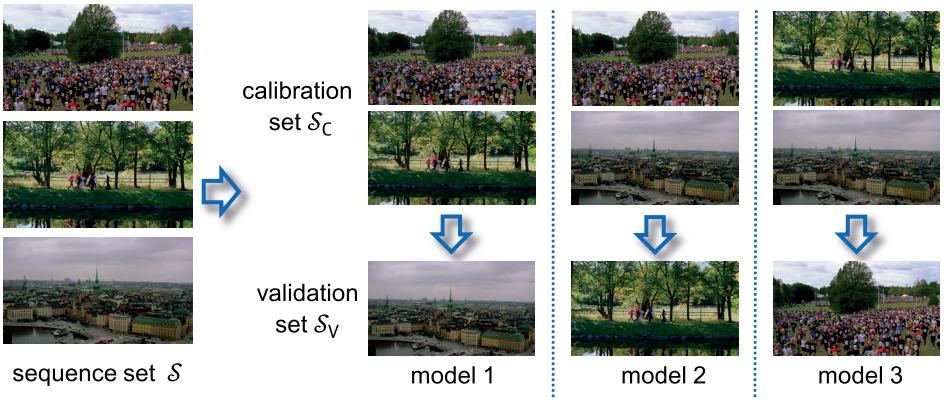

Figure 7.1.: Leave-one-out cross validation for $N = 3$ video sequences: splitting up in calibration set $\mathcal{S}_C$ and validation set $\mathcal{S}_V$, followed by the calibration of three different models

pecially considering modern multi-core processors allowing the simultaneous calibration of multiple models. Leave-one-out cross validation is commonly used for the validation of models based on data analysis methods [197, 298] and therefore the leave-one-out cross validation is also the cross validation method used in this thesis.

K-fold cross validation *K-fold cross validation* can be considered as a compromise between the hold-out and leave-one-out cross validation and was introduced by Geisser [72]. It is a method that applies a partial splitting strategy by partitioning $\mathcal{S}$ into K subsets $\mathcal{S}_k$ of equal size $|\mathcal{S}_k| = N/K$ for $k = 1, \ldots, K$, with $K \ll N$, $\mathcal{S} = \bigcap_{k=1}^{K} \mathcal{S}_k$ and $\bigcup_{k=1}^{K} \mathcal{S}_k = \varnothing$. This results in $N_C = N - N/K$ sequences for calibration, N/K sequences for validation with $N_V = N/K$ and overall K different models. The k-fold cross validation can be considered as a reduced leave-one-out cross validation with less computational complexity as only $K \ll N$ models need to be determined. Also for $K = N$ the k-fold cross validation is equivalent to the leave-one-out cross validation. Considering the k-fold cross validation as an estimator for the prediction error, the bias will be larger due to the smaller calibration set with $N/K < N$ and the variance is increased to the additional variability introduced by the partitioning into K subsets. Results by Breiman and Spector [26], however, suggest that k-fold cross validation is better suited for model selection than the leave-one-out cross validation.

One assumption in cross validation is that each of the N samples is an independent and identically distributed sample from the general population. In video quality assessment, however, it is common that although we have N different video sequences $\underline{\mathbf{S}}_n$, we only have I different sources. The N different video sequences are usually generated by processing each of the I video contents with $J = N/I$ different parameters e.g. different bitrates during encoding. Clearly, all video sequences $\underline{\mathbf{S}}_{ij}$ resulting from the processing of the content i with different parameters j will share common properties. Hence, the $N = IJ$

video sequences $\underline{\mathbf{S}}_n$ can no longer be considered independent and identically distributed. Denoting the set of sequences with the same content i as $\mathcal{S}_i = \{\underline{\mathbf{S}}_{i1}, \underline{\mathbf{S}}_{i2}, \ldots, \underline{\mathbf{S}}_{iJ}\}$, the set of all sequences is then given by $\mathcal{S} = \bigcap_{i=1}^{I} \mathcal{S}_i$ and the sets with different content are disjunct $\bigcup_{i=1}^{I} \mathcal{S}_i = \varnothing$. Assuming a leave-one-out cross validation, we thus have both $IJ - J$ calibration sets $|\mathcal{S}_{C,i}| = IJ - J$ and validation sets $|\mathcal{S}_{V,i}| = J$, where $\mathcal{S}_{V,i} = \mathcal{S}_i$ and $\mathcal{S}_{C,i} = \mathcal{S} \setminus \mathcal{S}_i$ for $i = 1, \ldots, I$. Hence, instead of a leave-one-out cross validation between N video sequences, we have a cross-validation between I sets of different content. One could argue that we therefore no longer apply a leave-one-out cross validation, but rather a k-fold cross validation. But unlike the premise of the k-fold cross validation, we are not splitting our data into larger subsets to reduce the number of calibration and validation sets, but rather due to the inherent structure of the data in our a samples and the requirements of the cross validation for independent and identically distributed data.

For further information about cross validation in the context of data analysis, I refer to Martens and Næs [197], and for a discussion of cross validation in general, I refer to the survey by Arlot and Celisse [10] and the references therein.

7.2. Model selection

Validation allows us to assess the predictive abilities of models based on different features or design methods. Clearly, we chose the model with the best predictive abilities. It is, however, possible that although the models have different properties, they still have the same or at least nearly the same prediction abilities. Thus we need criteria how to select the most suitable model not only based on its predictive abilities, but also on its structure.

Before discussing the selection criteria, it is useful to briefly review the relationship between the overall model complexity represented by the number of a model's parameters and the prediction error: on the one hand, if the model is too simple, it is not able to explain the systematic structure of the latent variables well enough, but on the other hand, if the model is too complex the noise increases due to the estimation of more parameters. This leads to *underfitting* and *overfitting* of the model on the opposite ends of the complexity scale, and an optimal choice of complexity for minimal prediction error in between these two extremes as illustrated schematically in Fig. 7.2 on the facing page [197]. Although it is counter intuitive, more parameters are therefore not necessarily better and consequently the aim should be to use as many parameters as necessary, but as few parameters as possible. Therefore for multiple models with equal predictive abilities, but different complexity, the model with the least complexity should be chosen.

This strategy of selecting the least complex model is known as the principle of *parsimony* and the aim is to choose the most *parsimonious* model, representing the simplest possible model from a set of different models fulfilling a given criterion equally well. The concept of parsimony is often referred to as *Ockham's razor*, but can be traced even further back [319]. In the context of factor or component analysis, the general notion of parsimony was first introduced by Thurstone in [320] suggesting that factor models with

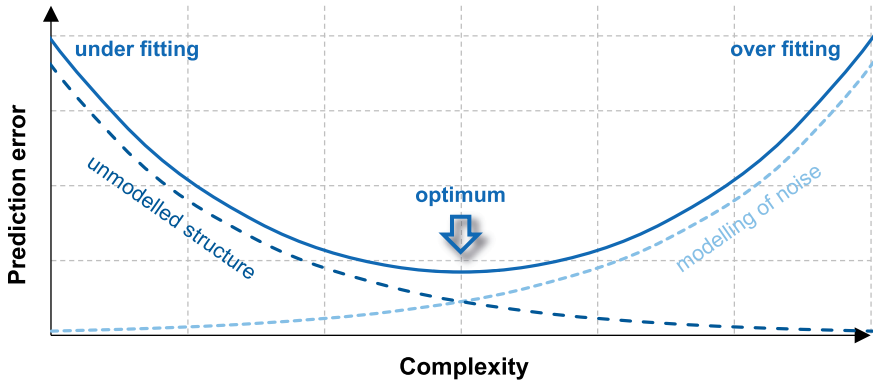

Figure 7.2.: Prediction error depending on the model complexity, leading to *underfitting* or *overfitting* on the extremes and an optimum in between (adapted from [197])

simple structure should be preferred. This was later formalised by Ferguson in [61], by explicitly discussing parsimony in the context of factor analysis and suggesting a parsimony measure based on the loadings of a factor model. In particular, Ferguson related parsimony to Shannon's definition of information [286], suggesting that the most parsimonious model is the model containing maximum information about the phenomena described by the model. Ferguson, however, did not provide a proof that a more parsimonious model provides better predictive abilities. Parsimony in model selection was therefore subsequently addressed by Seasholtz and Kowalski for component models based on PCR and PLS in [281]. Seasholtz and Kowalski have shown that for linear models and a given prediction error function, the model with fewer parameters will on average have a smaller prediction error asymptotically. Hence, the selection of the most parsimonious model can be justified by the minimisation of the prediction error.

Parsimony in this thesis is focused on the number of components as the main model parameter in the different models and thus for models with equal predictive abilities, the model with fewer components will be preferred. As the components are a representation of the latent variables, fewer components for the same predictive abilities represent a more compact representation of the structure inherent in the three-way feature array $\underline{\mathbf{X}}$ and, depending on the used data analysis method, the relationship of the features to the visual quality vector $\mathbf{y}$. Additionally, the features themselves can also be considered as model parameters and therefore if some features do not have any noticeable influence on the predictive abilities of a model, these features may be omitted from the model building process in order to gain a more parsimonious model. Assuming, however, that usually only features with a known or at least strongly suspected influence on the visual quality are selected for the model building process, it may not always be possible to reduce the models' complexity by using fewer features and thus the number of components is therefore the main parameter in the optimisation for the most parsimonious model.

7.3. Component selection

In discussing the different two-way and multi-way data analysis methods, the number of R components to be extracted was considered to be a predetermined parameter. Assuming we have a three-way feature array $\underline{\mathbf{X}} \in \mathbb{R}^{N \times M \times T}$ and usually $M \ll T$, we can at most extract $R = M$ components and we aim at a model with $R < M$ components, describing the latent variables within $\underline{\mathbf{X}}$

Our main goal, however, is to select those R components that provide the best prediction abilities for unknown video sequences not included in the calibration set $\mathcal{S}_C$. This selection is usually done using leave-one-out cross valuation to assess the prediction error of the model for each component r with $r = 1, \ldots, R, \ldots, M$. One advantage of using a cross validation approach is its universal applicability, regardless of the used data analysis method. Moreover, it can be used for two-way and three-way data analysis methods alike. The prediction error is usually defined as the mean squared error between $\widehat{y}$ and y, resulting in the *mean squared error of prediction (MSEP)* as

$$MSEP = \frac{1}{N} \sum_{n=1}^{N} (y_n - \widehat{y}_n)^2, \tag{7.3.1}$$

where for the prediction $\widehat{y}_n$ the model determined with $\mathcal{S}_C \setminus \underline{\mathbf{S}}_n$ is used. This is repeated for all r, resulting in M $MSEP_r$ that describe the prediction error for the model with r components. Note, that in literature e.g. Martens and Næs [197] the MSEP in cross validation is sometimes denoted as MSECV and MSEP is used to denote the MSE for the prediction error between the model and a separate validation set $\mathcal{S}_P$. Similarly, the *mean squared error of calibration (MSEC)* can be determined, providing information of the fit of the model with respect to the calibration data without cross validation where $\mathcal{S} = \mathcal{S}_C = \mathcal{S}_V$. Accounting for the degrees of freedom in model building, the MSEC can be expressed as

$$MSEC = \frac{1}{N - (r+1)} \sum_{n=1}^{N} (y_n - \widehat{y}_n)^2, \tag{7.3.2}$$

where $r+1$ is the term suggested by Martens and Næs [197] to compensate for the degrees of freedom in PLS that is also used for other methods in this thesis for convenience. Often the *root mean squared error (RMSE)* is used as alternative error metric and then the equivalent errors to the MSEC and MSEP are denoted as RMSEC and RMSEP, respectively.

The number of components R is then determined as the index r of the component that results in the (first) minimum of the MSEP and this is often determined visually using a so-called *PRESS plot* that plots the MSEP depending on the number of components r, where PRESS stands for prediction sum of squares (PRESS) as suggested in the leave-one-out cross validation proposal by Allen [2]. Note that the only difference between the MSEP and the PRESS is the averaging overall sequences N. A PRESS plot is illustrated

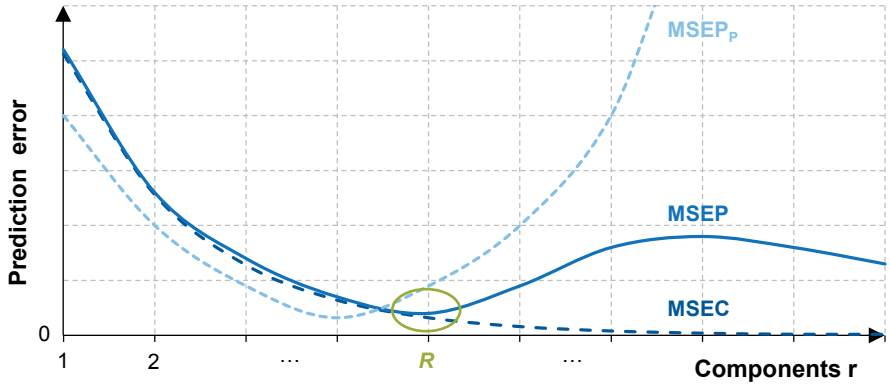

Figure 7.3.: PRESS plot: prediction error *MSEP* for leave-one-out cross validation and calibration error *MSEC* depending on the selected number of components r. Additionally the prediction error $MSEP_P$ for a separate set $\mathcal{S}_P$ (adapted from [197])

schematically in Fig. 7.3 and we can observe three typical properties of a PRESS plot for a prediction model: firstly, the MSEC describing the model's fit to the calibration set $\mathcal{S}_C$ approaches asymptotically zero for an increasing number of components. This is not surprising, as with each component the model approximates the sequences within $\mathcal{S}_C$ better. Secondly, the MSEP describing the prediction error of the cross validation shows a bath-tub like behaviour by first decreasing, reaching a minimum an then starting to increase again. Of particular interest is the minimum MSEP as the corresponding component r is the component number R we are looking for that is minimising the prediction error. The overall behaviour is explained by the increasingly better structural model with each component and therefore better prediction abilities for unknown video sequences until the R-th component. But from the $R+1$ component the model starts to be over-fitted to the calibration set $\mathcal{S}_C$, mostly representing the noise in $\mathcal{S}_C$ and thus the prediction abilities of the model diminishes again. Lastly, we can observe that the prediction error $MSEP_P$ for a separate set of video sequences $\mathcal{S}_P$ is higher than the *MSEP* for the prediction from the cross validation. This can be explained by the fact that the cross validation is just an estimator of the true prediction error and tends to underestimate the prediction error. This was shown empirically for small data sets by Martens and Dardenne [199]. Wold [360] described this visual approach using the PRESS-plot in his R_{Wold}-criterion as $R_{Wold} = MSEP(r+1)/MSEP(r)$, where $R \equiv r$ for the first r with $R_{Wold} \geq 1$ or as suggested by Krzanowski [163] for the first r with $R_{Wold} > 0.9$. This was extended by Osten [230] to include an F-test in the decision to find the best r, but van der Voet [331] argued that an F-test is not suitable in this context.

Other criteria for component selection without cross validation include the MSEC with leverage correction [197], using the size of the eigenvalues of the variation explained by each of the r components [348] or a randomisation approach where a test statistic based

on the calibration set is compared to the results of the same test statistic for multiple permuted versions of the calibration set [348]. Additionally, Næs and Helland [222] propose to classify components into components with strong and weak relevance for the prediction model: if the scores **T** are spanned by the eigenvectors of the covariance matrix of **X** and the regression weights with respect to the scores **q** are in the space spanned by the columns of **T**, the components are relevant, otherwise they are not.

7.4. Feature selection

In the building process of a visual quality prediction model, usually only features in the three-way feature array $\underline{\mathbf{X}}$ are included that are known or at least strongly suspected to have a significant influence on the visual quality. But even with this a-priori selection of features, it may very well be that some of the features may have only a negligible influence on the quality prediction and in order to gain a more parsimonious prediction model such features should be excluded from the model. One option to determine these features is a separate leave-one-out cross validation for each of the M features in addition to the leave-one-out cross validation for the component selection, enabling us to examine the influence of the individual features on the prediction error. After identifying features with no or only small influence on the prediction error, the model building process is then repeated without these features. Considering, however, that we already perform a leave-one-out cross validation with N models for determining the components R this would lead to NM different models, with $NM \gg N$ if we assume that a sufficiently large number of M features are used. In order to avoid building these additional models, other methods allowing us to directly analyse the contribution of the individual features to the predictive abilities of the model are needed. In this section, I therefore briefly discuss two methods for determining the influence of individual features on the prediction and thus enabling us to select only useful features.

Jackknifing *Jackknifing* is a method closely related to cross validation. The idea is to compare the regression weights of the features gained in each of the N bilinear models in the leave-one-out cross validation with the regression weights $\widehat{\mathbf{b}} \in \mathbb{R}^{M\times 1}$ gained for the model without cross validation i.e. $\mathcal{S}_C = \mathcal{S}_V$. The jackknife was adapted by Martens and Martens [200] for the use in two-way data analysis and is based on the jackknife estimation of bias and variance proposed by Quenouille [261] and Tukey [327], respectively. For the prediction model with the n-th video sequence $\underline{\mathbf{S}}_n$ left out in the calibration set $\mathcal{S}_C$, the regression weights are denoted as $\widehat{\mathbf{b}}_n \in \mathbb{R}^{M\times 1}$ and the estimated variance $\widehat{\boldsymbol{\sigma}}^2 \in \mathbb{R}^{M\times 1}$ for all features is then given as

$$\widehat{\boldsymbol{\sigma}}^2 = \frac{N-1}{N}\sum_{n=1}^{N}(\widehat{\mathbf{b}} - \widehat{\mathbf{b}}_n)^2, \tag{7.4.1}$$

where the m-th element $\widehat{\sigma}_m^2$ of $\widehat{\boldsymbol{\sigma}}^2$ corresponds to the estimated variance of the m-th feature. Using $\widehat{\sigma}_m^2$, a standard t-test can then be performed for each regression weight $\widehat{b}_m$ representing the influence of the m-th feature in the prediction model with the null hypothesis that $\widehat{b}_m = 0$, where Martens and Martens [200] suggest to use a significance level of $p < 0.1$. If the results for one or more of the features is above the significance level, the feature with the highest p-value is removed and the model building process is repeated with $M - 1$ features. This whole procedure is repeated until all remaining features are below the chosen significance level [3]. Note, that if there is a large number of insignificant features, this iterative process may lead to an increased number of built models to be built, similar as for the basic cross-validation approach discussed before. Instead of using the regression weights $\widehat{\mathbf{b}}$, also loadings $\mathbf{P}$ or loading weights $\mathbf{W}$ can be used to identify features insignificant for the prediction, but a correction for rotational ambiguity in bilinear models needs to be applied and the jackknifing must be performed for each component separately [200]. The jackknifing of loadings $\mathbf{P}$ or loading weights $\mathbf{W}$ can be extended to multi-way arrays, but the jackknife can only be performed for each mode separately as discussed by Riu and Bro [271]. For more information on the jackknife in the context of data analysis and in general, I refer to Martens and Martens [200] and Efron [51], respectively.

Leverage *Leverage* is a concept to describe the influence of individual features on the prediction model, where features with a high leverage have strong influence and features with a low leverage have a weak influence on the prediction model. It therefore provides a measure of uniqueness of the individual features compared to all other features [221]. The concept of leverage as originally introduced in the context of MLR by Hoaglin and Welsch [88] is equivalent to the *squared Mahalanobis distance* [193] upon a scaling factor and offset. For a two-way array $\mathbf{A} \in \mathbb{R}^{I \times J}$, where the I rows represent different objects, the squared (sample) Mahalanobis distance Δ^2 for the i-th object represent by the i-th row vector $\mathbf{a}_i \in \mathbb{R}^{1 \times J}$ can be expressed as

$$\Delta_i^2 = (\mathbf{a}_i - \overline{\mathbf{a}})^\top (\mathbf{A}^\top \mathbf{A})^{-\mathbf{1}} (\mathbf{a}_i - \overline{\mathbf{a}}), \tag{7.4.2}$$

where $(\mathbf{A}^\top \mathbf{A})^{-\mathbf{1}}$ represents the covariance matrix of $\mathbf{A}$ and the row vector $\overline{\mathbf{a}} \in \mathbb{R}^{1 \times J}$ the average over all I objects in $\mathbf{A}$. Clearly, for uncorrelated variables the covariance matrix is equal to the identity matrix and then the Mahalanobis distance is equal to the Euclidean distance to the calibration centre.

The squared Mahalanobis distance can therefore be interpreted as the Euclidean distance weighted by the inverse covariance matrix of the samples. It takes into account the influence of the correlation between the variables represented by the covariance matrix and thus provides an estimation of the probability distribution of the objects. The Mahalanobis distance is thus the distance between a object and the calibration centre weighted with this probability distribution. Hence, a small distance indicates an object with less importance and thus small leverage, as it is close to the calibration centre and the probability of observing an object with this variable combination is high. On the contrary, a large distance indicates an object with high importance and thus large leverage, as it is far from the

calibration centre and the probability of observing an object with this variable combination is low. In the later case, however, this could also be an indication for an outlier, and not a object with high leverage.

Considering that we are interested in the leverage $\mathbf{h} \in \mathbb{R}^{M\times 1}$ of the features in order to perform a feature selection, the squared Mahalabonis distance of the loadings $\mathbf{P} \in \mathbb{R}^{M\times R}$ generated by a two-way data analysis for given number of r components is then provided as

$$\mathbf{h} = \operatorname{diag}\left[\mathbf{P}(\mathbf{P}^\top\mathbf{P})^{-\mathbf{1}}\mathbf{P}^\top\right], \tag{7.4.3}$$

where loadings are the objects and by definition offset-free. The m-th element h_m of $\mathbf{h}$ represents then the leverage of the m-th feature with $0 \leq h_m \leq 1$ [298]. If the loadings are not directly available, the leverage of the features can also be determined with the loading weights $\mathbf{W}$ and by replacing the loadings $\mathbf{P}$ with the scores $\mathbf{T}$ it is also possible to determine the leverage of the video sequences $\underline{\mathbf{S}}_n$ that may have little or large influence on the model building process. For multi-way arrays, the leverage is determined for each mode separately. Once features with low leverage are identified, they are removed from the three-way feature array $\underline{\mathbf{X}}$ and the model building process is repeated. As no cross validation is strictly necessary for the leverage determination, this approach is therefore also suitable for component models without regression. For a more information about the Mahalanobis distance, the concept of leverage in regression and in multivariate calibration, I refer to Maesschalck et al. [192], Cook and Weisberg [46] and Martens and Næs [197], respectively.

Part III.

Design of Video Quality Metrics

8. Designing Video Quality Metrics

In the previous chapters, I presented different data analysis methods that can be used to design video quality metrics. But in order to design such metrics, we need objective features for the three-way feature array $\underline{\mathbf{X}}$ in addition to the visual quality vector $\mathbf{y}$ gained in subjective testing.

Since the focus of this thesis is on the design of video quality metrics with multi-way data analysis in general and not on the development of a particular metric itself, it is not sensible to design a video quality metric highly optimised for a certain application. Still, we need to assess if multi-way data analysis provides an advantage in the design of video quality metrics and thus need metrics designed with the previously discussed methods in order to evaluate the suitability of the multi-way data analysis approach to the data driven design of video quality metrics.

In this chapter, I therefore provide two simple examples for no-reference video quality metrics, representing the two main categories in the engineering approach to video quality metrics: one bitstream-based and one pixel-based video quality metric. The first example describes how a video quality metric can be designed with *H.264/AVC bitstream-based* no-reference features, suitable for the visual quality estimation of H.264/AVC encoded video sequences. The second example describes a more universal approach, utilizing only pixel-based no-reference features and is thus also suitable for other coding technologies. As the focus of this thesis is on distortions introduced by coding artefacts, the features are selected with respect to this criterion and for other distortion types, in particular transmission errors or packet loss, other features or at least other extraction methods for some features may be necessary.

Metric framework Considering the data driven data analysis approach more generally as a method to build prediction models based on features representing the video sequences' properties, we can regard the structure of the resulting metrics as a framework consisting of a feature processing part and a separate prediction part as illustrated in Fig. 8.1 on the following page. The feature processing part describes the feature extraction and subsequent processing of the video sequences, and the prediction part then utilises the extracted features for predicting the visual quality. In the examples in this chapter, the feature processing is represented by the bitstream-based and pixel-based metric extracting the corresponding features, and the prediction part is represented by the different models generated with two-way and multi-way data analysis.

Using this module-based approach within a metric instead of a monolithic metric, we can easily utilise additional features and the corresponding extraction algorithms in the model

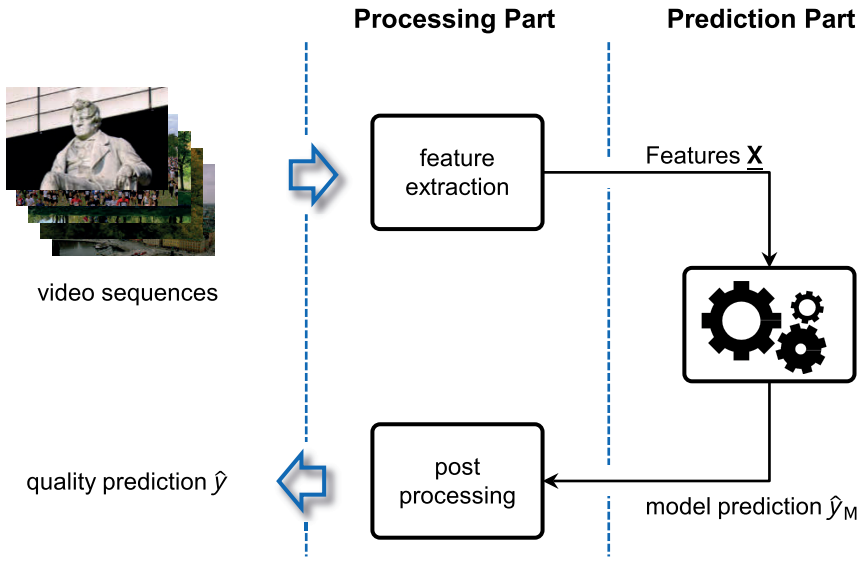

Figure 8.1.: Framework for video quality metrics: feature dependent processing part and feature independent prediction part

building process with data analysis. Thus even though the examples in this chapter focus only on a specific set of pixel-based or bitstream-based features, the same basic structure can be used for other features or even feature categories. Moreover, we can also include additional pre- or post-processing steps if necessary without affecting the prediction part. Alternatively, the data analysis based prediction models could also be replaced by models based on other principles or concepts.

8.1. Example I: H.264/AVC bitstream-based no-reference metric

In the first example, we use certain properties of the H.264/AVC bitstream for the prediction. No-reference in this context means that the video sequence itself will not be decoded, but that the entropy encoding for transmission has been reversed and that we have therefore direct access to the H.264/AVC bitstream.

An obvious limitation is that a metric based on H.264/AVC bitstream features will only be able to assess the visual quality of H.264/AVC encoded video sequences. H.264/AVC, however, is used in ever more applications from IPTV to HDTV and sometimes even instead of MPEG-2 for SDTV. Hence, we address a sufficiently large number of application areas, even if we only focus on H.264/AVC coding technology The feature and

overall design discussed in this section was also used to design the metrics presented in [142, 143, 146, 153]

8.1.1. The H.264/AVC standard

Before describing the extracted features, I will shortly provide a quick overview of the H.264/AVC video coding standard with a special focus on those properties that I will utilize in the design template. For more detailed information about H.264/AVC, I refer the interested reader to the abundant literature on this topic e.g. [315, 347] and of course the standard itself.

Commonly known as *H.264/AVC*, the recommendation ITU-T H.264 | international standard ISO/IEC 14496-10 - MPEG-4 Part 10, Advanced Video Coding [97, 126], is the result of the joint effort of ITU-T and ISO/IEC to create a more efficient successor especially for the then future application area of HDTV to both the ISO/IEC standards MPEG-2, Part 2 and MPEG-4, Part 2 [96, 98] and the ITU-T recommendations H.262 and H.263 [115, 117]. The overall aim was to achieve a reduction of the data rate by 50% compared to its predecessors while maintaining the same visual quality. Since the recommendation standard went into force in 2003, it has become the predominantly used video coding standard for new, non-legacy applications.

Overview H.264/AVC follows the model of hybrid coding already used by its predecessors. The basic principle behind this approach is illustrated in Fig. 8.2 on the next page. It combines a quantised integer transform coding with a motion compensated prediction. The encoder includes a decoder in the encoding loop that decodes the previous frame. It then performs a motion estimation and compensation of the frame with respect to the previous frame, before subtracting the motion compensated prediction from it. Hence, we only have to use the entropy coding to encode the transformed and quantised prediction error between the current frame and its processor for transmission, the so-called residual error, thereby reducing the transmitted information significantly. Additionally, we also have to transmit the motion vectors used for the motion compensation, but this is usually significantly less data compared the size of non-motion-compensated frames, especially considering that the motion vector themselves can also be predicted from their preceding motion vectors.

Obviously, this is only possible, if already at least one frame has been transmitted. Therefore we can distinguish in H.264/AVC two basic frame types: *intra* and *inter*. Intra frames, also called I-frames, may only use information in the current frame for encoding, whereas inter frames may also use information from other frames. Depending on the inter frame type, only previous or both previous and successive frames may be used for P-frames and B-frames, respectively. Multiple frames are then grouped into a group of pictures (GOP) that starts with an I-frame and can be considered the smallest independently decodable unit of a video sequence.

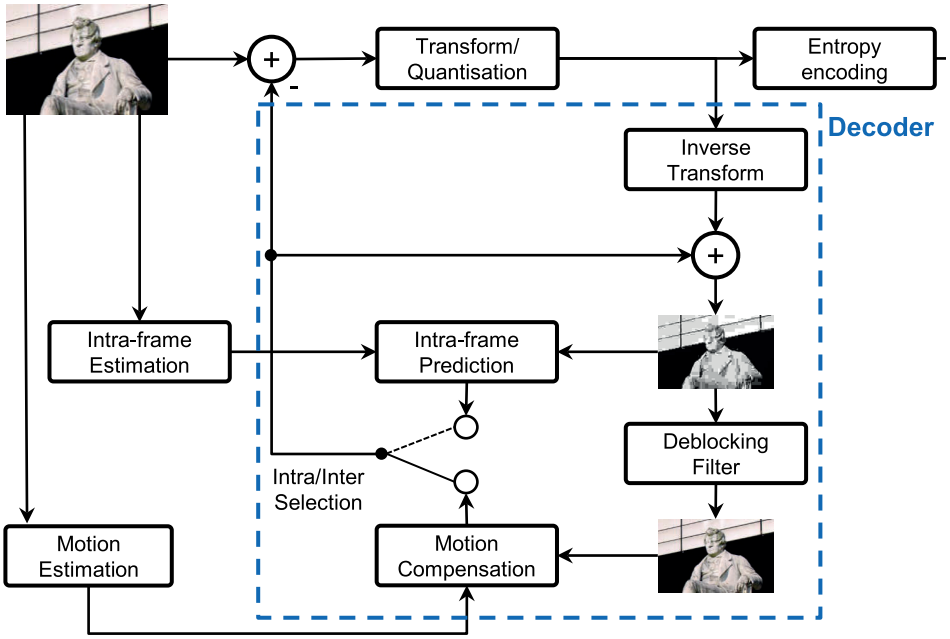

Figure 8.2.: Simplified overview of the H.264/AVC encoding process

Requirements on the decoder with respect to their support of certain H.264/AVC features are described in profiles and levels describe the required computational performance for certain resolutions and frame rates.

Frame structure Each frame is divided into 16 $\times$ 16 macroblocks that are the basic unit for both motion compensation and transform coding. They can be further portioned into 8 $\times$ 8 and 4 $\times$ 4 submacroblocks for intra coded macroblocks and 16 $\times$ 8, 8 $\times$ 16, 8 $\times$ 8, 8 $\times$ 4, 4 $\times$ 8 and 4 $\times$ 4 submacroblocks for inter coded macroblocks. Note, that depending on the frame type, different macroblocks and their corresponding prediction references may be used in the same frame e.g. B-frames may combine intra and inter coded macro blocks. Moreover, H.264/AVC also allows to flag macroblocks to be *skipped*, if they contain no residual error i.e. no change occurred and their motion can be effectively predicted from neighbouring macroblocks. In Fig. 8.3 on the facing page an example of a frame's possible subdivisions are shown. Note how the submacroblock sizes adapt to the details in the frame: for homogeneous areas with few details e.g. the sky in the background, larger submacroblocks or even the macroblocks themselves without any subdivision at all are used, whereas for areas with more details e.g. the building in the foreground, smaller subdivisions are used. Consequently, we will need less data to describe the areas with less information and can invest more bits into the details.

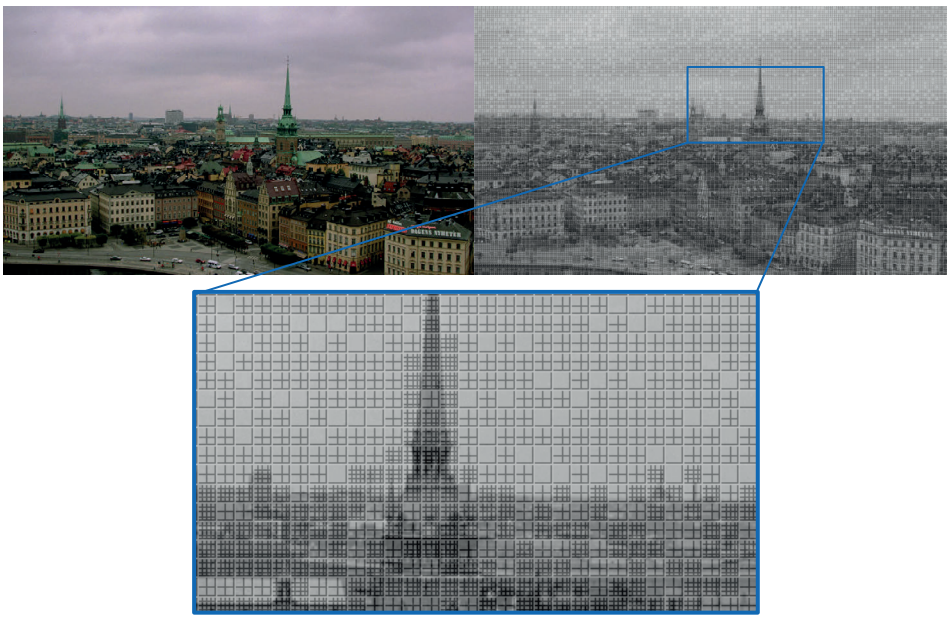

Figure 8.3.: Example of H.264/AVC macroblock structure and how the submacroblocks take the details in the frame into consideration (derived from [153])

Additionally, each frame can also be subdivided into multiple slices, which in turn contain multiple macroblocks. The advantage is that each slice can be decoded independently of the other slices, thus increasing the error resiliency. Each slice itself can be of an intra or inter type. In most real-life applications, however, each frame is usually represented by exactly one slice. Hence, the term slice and frame can often be used synonymously.

Bitstream structure In order to extract features directly from the bitstream, we need to consider its different conceptual layers as shown in Fig. 8.4 on the next page.

The Network Abstraction Layer (NAL) represents the top layer from a bitstream perspective. It allows a simple and effective adaptation of the H.264/AVC bitstream to different transport networks and consists of packets, the NAL Units (NALU). We can distinguish between two different types of NALUs: Video Coding Layer (VCL) and non Video Coding Layer (non-VCL). The non-VCL NALUs contain information about the video transported in a VCL NALU. The sequence parameter sets (SPS) describe parameters concerning the complete video sequence, e.g. profile, and the picture parameter sets (PPS) describe decoding properties of one or more slices within the video sequence. Optional non-VCL types contain supplemental enhancement information (SEI) NALU or the mapping of the slices to the frames in an access unit (AU) NALU. As shown in Fig. 8.4 on the following page, each PPS refers to one SPS.

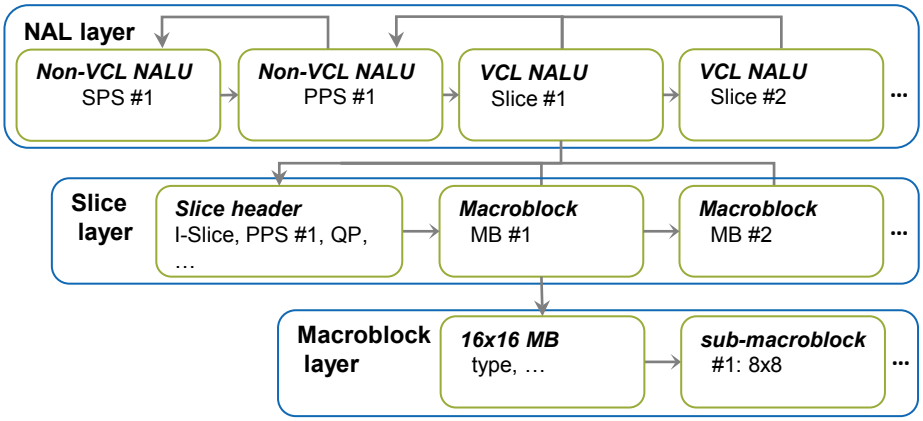

Figure 8.4.: H.264/AVC bitstream and its different conceptual layers (derived from [153])

Each VCL NALUs contains a slice, which leads us to the next layer, the slice layer. Here we find in the slice header parameters valid for the complete slice e.g. slice type or start quantisation point. Additionally, the VCL NALU contains all macroblocks of the corresponding slice. This brings us to the last layer, the macroblock layer. Here we can obtain information about the individual macroblocks, in particular their segmentation, their motion vectors and any deviation from the overall quantisation point of the slice. Note, that often each frame only consists of one slice and therefore a VCL NALU corresponds directly to one frame.

8.1.2. Extracted bitstream features

After this short introduction into the H.264/AVC standard, I present in this section the features of the H.264/AVC that will be used in the example metric. All in all, 17 different features are extracted from the bitstream. They can be classified into *slice parameters* and *macroblock parameters*. The first category describes features we can extract from the slice layer and describe properties of the complete slice, whereas the second category describe features we derive from the individual macroblocks in the macroblock layer.

Slice parameters In the slice layer, the following features are extracted:

- **Type** describing if the slice is an I-, P-, or B-slice, or considering that in most cases a slice is equivalent to a frame, if the current frame is an I-, P-, or B-frame.
- **Bitrate** describing the bitrate in kBit needed for coding the slice i.e. the size of the VCL NALU containing the slice.
- **QP** describing the initial quantisation point of the slice from the slice header.

Macroblock parameters In the macroblock layer, we collect information about the macroblocks in the corresponding slice and then pool them per slice. The collected features are:

- **Intra** percentage of the intra macroblocks in a slice.
- **Inter** percentage of the inter macroblocks in a slice.
- **Skip** percentage of the skip macroblocks in a slice.
- **I16x16** percentage of all intra macroblocks in a slice that are 16×16 macroblocks.
- **I8x8** percentage of all intra macroblocks in a slice that are subdivided in 8×8 submacroblocks.
- **I4x4** percentage of all intra macroblocks in a slice that are subdivided in 4×4 intra type submacroblocks.
- **P16x16** percentage of all inter makroblocks in a slice that are subdivided in 16×16 inter type macroblocks.
- **P8x8** percentage of inter type submacroblocks that were subdivided into 16×8, 8×16, 8×8 or smaller submacroblocks. This percentage is equal to the difference between the percentage of all inter macroblocks and the percentage of 16×16 submacroblocks.
- **P4x4** percentage of inter type 8×8 submacroblocks that were subdivided into 8×4, 4×8 and 4×4 submacroblocks. This percentage is equal to the difference between the percentage of all 8×8 submacroblocks and the percentage of smaller submacroblocks.
- **MVI** is the average motion vector length over all macroblocks in the slice. The motion vector can be determined for each macroblock *i* by the prediction of the motion vector $\mathbf{mv}_{pred,i}$ and its prediction error $\mathbf{mv}_{d,i}$.

$$\mathbf{mv}_i = \mathbf{mv}_{pred,i} + \mathbf{mv}_{d,i}. \tag{8.1.1}$$

  The average motion vector length over all macroblocks *I* in a slice is then given by:

$$\mathrm{MVI} = \sum_{i=1}^{I} \|\mathbf{mv}_i\|. \tag{8.1.2}$$

- $\mathbf{MVI_{Max}}$ is the maximum motion vector length over all macroblocks in the slice.
- **MVd** is the average motion vector difference between predicted motion vector and actual motion vector over all macroblocks in the slice.

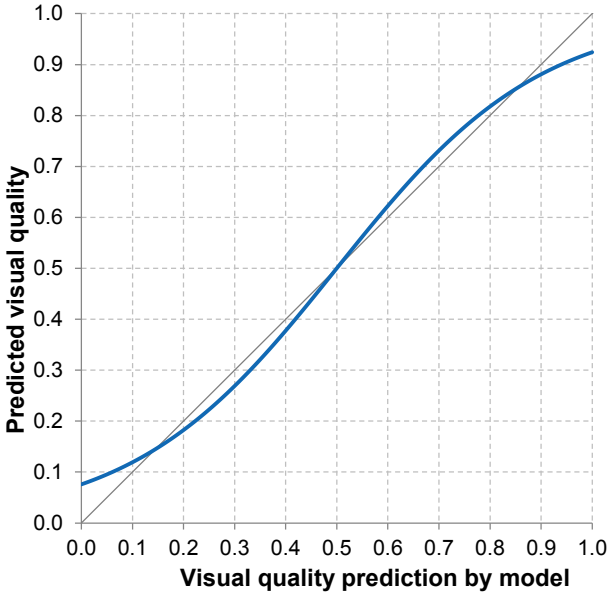

Figure 8.5.: Sigmoid correction of the visual quality $\widehat{y}_M$ as provided by the prediction model

- $\mathbf{MVd_{Max}}$ is the maximum motion vector difference between predicted motion vector and actual motion vector over all macroblocks in the slice.
- **QPd** is the average deviation of the QP over all macroblocks in a slice.

An example of the variation within selected bistream features across different visual quality levels is shown in Fig. 8.6 on the next page for the video sequence *Foreman* from the IT-IST data set [23–25] at four different visual quality levels. Note, that we do not aim in the data analysis approach to interpret the features directly with respect to their influence on the visual quality, but are rather interested if the features exhibit some variation between different visual quality levels.

Due to limitations of the modified H.264/AVC JM decoder used for the bitstream-based feature extraction, the proposed template for a H.264/AVC bitstream-based no-reference metric is suitable for progressive video sequences without transmission errors. Additionally, the encoded video sequences are only allowed to have one slice per frame and the bitstream has to be in the Annex B format.

8.1.3. Post processing

In order to take into account the biases encountered in the subjective ratings as discussed in Section 2.5.2 that lead to an avoidance of the limits of the used quality scale the example metric includes a sigmoid correction emulating this behaviour, particularly the effect of

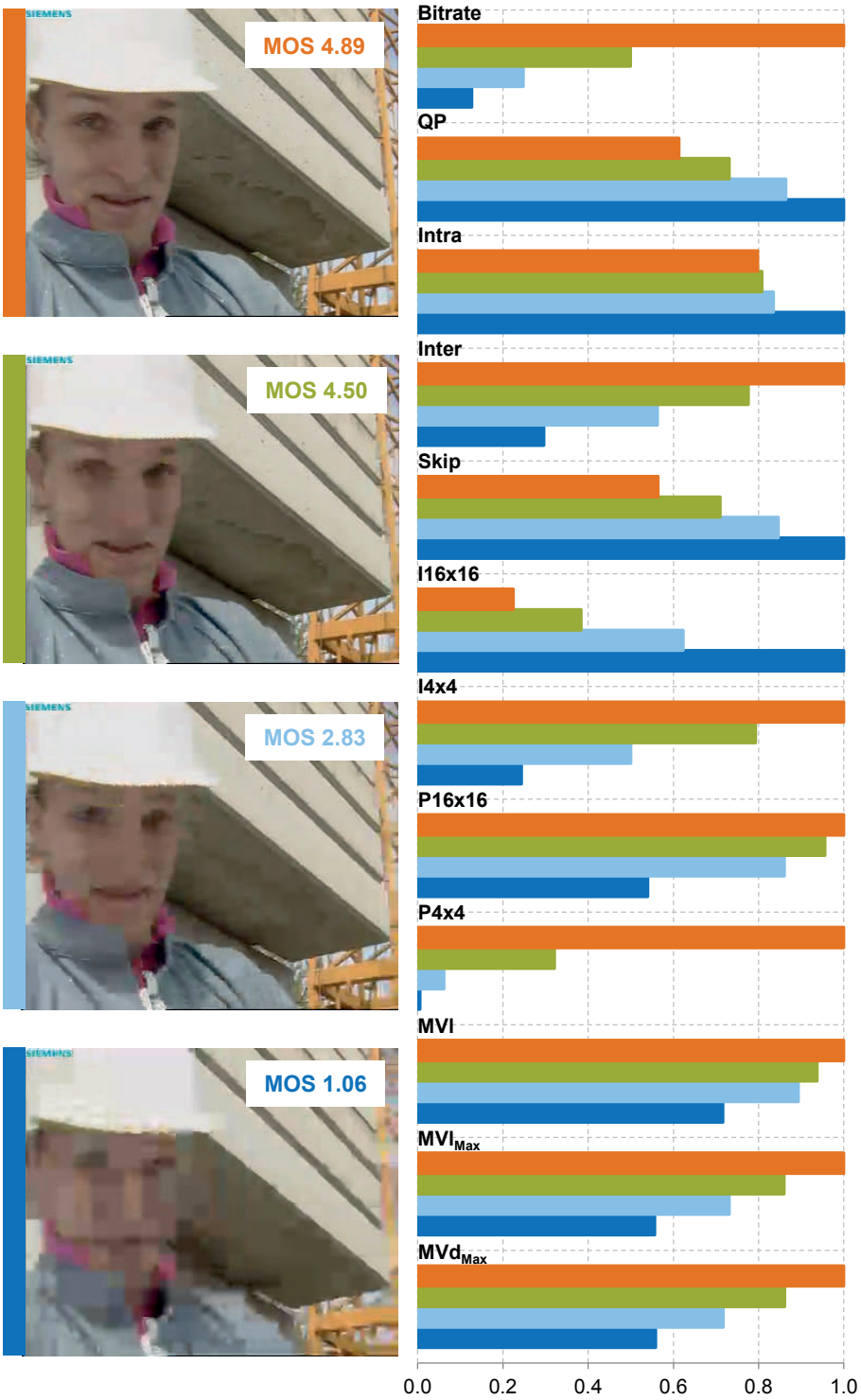

Figure 8.6.: Example of the variation within selected bitstream features across different quality levels: *Foreman* at different visual quality from *bad* with a MOS of 1.06 to *excellent* with a MOS of 4.89. For comparability all features have been normalised to their maximum value

the contraction bias, in the post processing of the visual quality $\widehat{y}_M$ as provided by the prediction model.

The sigmoid correction for the visual quality prediction $\widehat{y}$ is given as

$$\widehat{y} = \frac{a}{\left(1 + e^{-\frac{(\widehat{y}_M - b)}{c}}\right)}, \tag{8.1.3}$$

where $\widehat{y}_M$ is the visual quality provided by the model and a, b and c are the parameters of the sigmoid function.

The values for the parameters were determined empirically and are set to $a = 1.0$, $b = 0.5$ and $c = 0.2$, leading to a mapping of $\widehat{y}_M$ to the visual quality $\widehat{y}$ that exhibits similar properties to results of subjective testing with respect to the utilisation of the visual quality scale. The resulting sigmoid function is shown in Fig. 8.5 on page 134. Note, that this function is not adapted to the actual data, but is rather a fixed part of the design template.

8.2. Example II: Pixel-based no-reference metric

Pixel-based video quality metrics allow us to overcome the main limitation of bitstream-based video quality metrics, their limitation to a certain coding technology or bitstream format. Therefore the second example describes a no-reference pixel-based video quality metric that uses only the decoded frames of the video sequences for the quality estimation. The example is based on the metrics presented in [145, 148, 224] and the work by Oelbaum [226].

8.2.1. Extracted pixel-based features

All in all, nine different pixel-based features are extracted from the decoded frames of the video sequences and will be described briefly in this section. The first four features are only extracted from a single frame and can be considered as intra-frame features: *bluriness*, *blockiness*, *spatial activity* and *ringing*. In contrast to the intra-frame features, the last five features are extracted from consecutive frames and can therefore be considered as inter-frame features: *fast motion*, *temporal predictability*, *edge continuity*, *motion continuity* and *colour continuity*. These were originally proposed in [226] by Oelbaum. Obviously, all features relying on multiple frames are only available if preceding or succeeding frames exist and can therefore not be determined for the first or last frame in a video sequence. The main focus in this section is on the basic principles behind the features and their extraction. If not stated otherwise, the features are only determined in the luma component of a frame. For an in-depth discussion of the individual features, I refer to the given references for each feature.

Bluriness The first feature we extract provides us with a measurement of the of the decoded frames. *Blur* can be defined as the attenuation of the spatial high frequency components in the spectrum. Thus the effect on the content of the frame is similar to applying a spatial low pass. In general, it can be caused both by fast motion, the so-called motion blur, if the objects or the camera move during the capture of a frame, but it can also occur if the captured frame is out of focus. More specifically in the context of video coding, it occurs often if during the encoding high frequency coefficients of the used transform e.g. DCT are either completely discarded or quantised so coarsely that they practically disappear.

For measuring bluriness, the no-reference blur estimation proposed by Marziliano et al. [202] is used. The basic idea is to measure the width of edges in the frame, assuming that sharp edges will be widened, if blur occurs. Algorithm 8.1 on the next page describes the blur detection for vertical bluriness. For horizontal blur, Algorithm 8.1 is similarly applied on horizontal edges and columns. Both horizontal and vertical bluriness are used in the example metric.

It has a lower bound of 0, indicating that only sharp edges appear and a theoretical upper bound only limited by the frame size. For uncompressed frames, the bluriness is

Algorithm 8.1: No-reference blur detection [202]

```
apply vertical edge detector to frame e.g. vertical Sobel filter
for each row in filtered frame do
    for each pixel in row do
        determine local extrema closest to edge
        width of edge = difference between position of extrema
        set blur for edge to width of edge
    end
end
set bluriness to average of blur over all edges
```

approximately 1 and for frames with compression artefacts, the bluriness is approximately 5 [224]. In order to account for natural occurring motion blur, the bluriness value is adjusted by scaling, if fast motion is detected between two consecutive frames.

Blockiness In contrast to bluriness, *blockiness* provides us a with a measure of edges introduced by coding artefacts in the frame. Current coding technologies mostly utilize block-based transforms e.g. DCT or integer transform, that apply the transformation and quantisation for each block independently. Due to this segmentation of the frames into marcroblocks and sometimes even further into submacroblocks, visible edges can occur on the border between two different macroblocks, leading to the so-called blocking artefacts. Blockiness is measured with the no-reference blocking artefact estimation proposed by Wang et al. [337]. It assumes that edges caused by blocking can be detected as peaks in the power spectrum of a frame after subtracting a smoothed version of the power spectrum. Algorithm 8.2 describes the blocking detection for vertical blockiness and for horizontal blocking, Algorithm 8.2 is applied on the horizontal difference between the columns in the frame. *blockiness*

The lower bound for the blockiness is 0 and, depending on coding technology, the blockiness can reach values of up to 2 or 25 for H.264/AVC and MPEG-2, respectively [224]. Although the integrated deblocking filter in H.264/AVC leads to an overall much smaller

Algorithm 8.2: No-reference blocking detection [337]

```
for each row in frame do
    calculate absolute difference between consecutive rows
end
estimate power spectrum P of difference frame
P_M = median filtered P
set blockiness to P − P_M
```

Algorithm 8.3: Spatial activity [116, Annex A]

```
   counter = 0
   I = pixel intensity
1  for each row in frame do
2      for each pixel i in row do
3          if sgn(I_i − I_{i−1}) ≠ sgn(I_{i−1} − I_{i−2}) then
4              increment counter
           end
       end
   end
5  set spatial activity to average of counter over all pixels
```

blockiness value for H.264/AVC encoded video sequences, blockiness can still be a valuable feature for the video quality estimation as we are not interested in the absolute value, but rather in the variation of the feature's value.

Spatial activity *Spatial activity* is a feature that allows us to measure the amount of details present in the texture. It is based on the texture analysis part in ITU-T J.144, Annex A [116], where as an indication of the amount of details the number of turning points per row or column for each frame is counted. Algorithm 8.3 describes the horizontal spatial activity measurement. The vertical spatial activity is determined similarly by replacing the rows with the columns of the frame.

The premise is, that with decreasing visual quality of a video sequence, the details and therefore the variation of the pixel values is reduced. Consequently, the number of turning points will also decrease. Assuming that details are less noticeable in scenes with a fast motion, we adjust the spatial activity similar to the bluriness.

Ringing *Ringing* is introduced by an insufficient representation of high spatial frequency components in the video frame, usually due to the quantisation of the transform coefficients in the encoded video. This corresponds to sub-sampling without low-pass filtering and therefore is band limiting the spatial frequency in the video frame, possibly cutting of some frequencies describing sharp edges or detailed textures.

In Algorithm 8.4 on the next page, a deringing filter is applied to the frame and then the difference between the filtered and the not filtered frame is calculated. Assuming that the deringing filter removed the ringing sufficiently well, this difference then represents the encountering ringing. The overall value of ringing for a frame is given by the percentage of pixels above the threshold that was determined empirically as 10.

Algorithm 8.4: Ringing [226], using the deringing filter from [96, Annex F]

```
   counter = 0
   I = pixel intensity
1  apply deringing filter to frame
2  calculate difference between frame and filtered frame
3  for each pixel in difference frame do
4  |  if |I_filtered difference frame − I_difference frame| > 10 then
5  |  |  increment counter
   |  end
   end
6  set ringing to counter / number of pixels
```

Fast motion *Fast motion* describes the amount of fast motion detected between two consecutive frames and is described in Algorithm 8.5. Unlike the features discussed so far that were extracted only from a single frame, fast motion is extracted from the difference between to frames.

Motion is considered to be fast if the length of either the vertical or horizontal motion vector between the current and the preceding frame is larger than the threshold of 4 pixel that was determined empirically. The assumption is that due to the spatio-temporal properties of the human visual system, fast motion may mask some of the distortions and therefore the amount of fast motion in a frame influences the visual quality. The overall amount of fast motion in a frame is then defined as the percentage of motion vectors classified as representing fast motion.

Temporal predictability *Temporal predictability* describes how well a frame can be described by its preceding frame. The assumption is that the transition between two consecutive frames in a video sequence without cuts is smooth. Hence, the current frame should be easily predictable from the previous frame.

Algorithm 8.5: Fast motion [226]

```
   counter = 0
1  V_P = motion vector field between current and previous frame
2  for each motion vector v_P in V_P do
3  |  if |v_P,x| ≤ 4 or |v_P,y| ≤ 4 then
4  |  |  increment counter
   |  end
   end
5  set fast motion to 1 − counter / number of motion vectors
```

Algorithm 8.6: Temporal predictability [226]

```
   counter = 0
1  perform motion compensation between current and previous frame
2  apply Gaussian low pass filter to current frame and the motion compensated version
   of the frame
3  apply median filter to frame and motion compensated frame
4  for each 8 × 8 block in frame do
5  |  calculate SAD between frame and motion compensated frame for all components
6  |  if SAD > 384 then
7  |  |  increment counter
   |  end
   end
8  set temporal predictability to 1 − counter / number of blocks in frame
```

Temporal predictability is then defined as the percentage of how many blocks are not noticeably different. If distortions due to coding artefacts occur, the number of noticeably different blocks will increase and therefore the temporal predictability will decrease. Algorithm 8.6 describes how the temporal predictability is determined. The Gaussian low pass and consecutive median filtering ensure that single pixels do not dominate the sum of absolute differences (SAD) between the current frame and its prediction gained by the motion compensation of the previous frame. The threshold for a noticeable difference of 384 was determined empirically and allows an average difference of 6 per pixel for a block size of 8×8.

Edge continuity The third inter frame feature is the *edge continuity*. It provides us with a measure of how much the structure of the video sequence changes between two frames. Once again, we first generate a motion compensated version of the current frame based on the previous frame. Then we calculate the EdgePSNR described in ITU-T J.144, Annex B [116] between both versions as described in Algorithm 8.7.

EdgePSNR assumes that the preservation of edges between a undistorted version and a distorted version of a frame can be a measure of its visual quality and therefore calculates the PSNR between the edges in both versions. For edge continuity we assume that the structure between two consecutive frames will not change significantly, if no scene change

Algorithm 8.7: Edge continuity using Edge PSNR [116, Annex B]

```
1  perform motion compensation between current and previous frame
2  calculate EdgePSNR between motion compensated and current frame
3  set edge continutiy to EdgePSNR
```

Algorithm 8.8: Motion continuity [226]

```
   counter = 0
1  V_P = motion vector field between current and previous frame
2  V_S = motion vector field between current and succeeding frame
3  for each motion vector v_P, v_S in V_P, V_S do
4  |   if |v_S,x - v_P,x| > 5 or |v_S,y - v_P,y| > 5 then
5  |   |   increment counter
   |   end
   end
6  set motion continuity to 1 − counter / number of motion vectors
```

occurs. Hence, the edges in the motion compensated version should not differ significantly from the current frame, unless introduced coding artefacts led to a discontinuity of the edges between both frames. The EdgePSNR can take values between 0 and 1, where 1 indicates that no changes in structure occurred between the two frames.

Motion continuity In the feature *motion continuity* we assess the motion trajectories between frames. The assumption is, that for natural video sequences the motion over multiple frames is smooth and that distortions introduced by coding artefacts result in less smooth motion trajectories. The motion continuity is determined with Algorithm 8.8, resulting in the percentage of smooth motion vectors as a descriptor for the motion continuity. The threshold of 5 pixels for the maximum deviation of the motion vectors in both spatial dimensions has been determined empirically.

Colour continuity The last inter-frame feature included in the design template is *color continuity*. This feature assumes that the colour histograms for two consecutive frames should be very similar, if no scene change occurs. Colour distortions e.g. colour bleeding introduced by coding artefacts, however, lead to different histograms.

Algorithm 8.9 determines the colour continuity, resulting in a value between 0 and 1. By correlating the histograms, colour continuity is relatively robust against gradual colour changes e.g. due to illumination changes in the scene, but for colour artefacts the resulting

Algorithm 8.9: Colour continuity [226]

```
   each histogram has 51 bins
1  perform motion compensation between current and previous frame
2  calculate colour histogram for current frame
3  calculate colour histogram for motion compensated frame
4  set colour continuity to linear correlation between both histograms
```

colour continuity values will be significantly lower. For 8 bit colour components, each bin corresponds to a range of five intensity levels, resulting in 51 bins for each colour component.

An example of the variation within selected pixel-based features across different visual quality levels is shown in Fig. 8.7 on the next page for the video sequence *Foreman* from the IT-IST data set [23–25] at four different visual quality levels. Note, that we do not aim in the data analysis approach to interpret the features directly with respect to their influence on the visual quality, but are rather interested if the features exhibit some variation between different visual quality levels.

8.2.2. Post processing

Unlike in the H.264/AVC bitstream-based example discussed in the previous section, the post processing in the pixel-based example metric not only includes a sigmoid correction, but also an adaptive correction step that corrects the quality prediction $\widehat{y}_M$ of the model depending on the actual video sequence and its sensitivity to coding artefacts. An overview of this post processing is shown in Fig. 8.8 on page 145. This correction step was first proposed for full reference and reduced reference metrics in [229] and [228], respectively, before being extended to no-reference metrics in [145] and [224].

In order to gain additional information about the sensitivity of the video sequence to coding artefacts, the sequence is first encoded to a lower quality by using an encoder with a sufficiently high, but still realistic quantisation point (QP) e.g. for H.264/AVC a QP of 45. Although we do not know the visual quality of the generated low quality version, we can safely assume that the visual quality will be significantly lower, regardless if the video originally had a high or low visual quality. By estimating the visual quality of the low quality version with the prediction model, we then get the predicted visual quality $\widehat{y}_{M,low}$, providing us with an indication about the sensitivity of the video to coding artefacts. This is then combined with additional information about the coding sensitivity of the video sequences used in the building of the model.

Let $\widehat{y}_{T,n}$ be the predicted visual quality of the n-th sequence in the training set during the building of the model. Then $\widehat{y}_{T,low,n}$ is the visual quality prediction of the low quality version of the n-th sequence, that was generated as described above. With $\overline{\widehat{y}}_{T,low}$ as the average of all $\widehat{y}_{T,low,n}$ in the training set and s_{low} the corresponding standard sample deviation, we first clip $\widehat{y}_{M,low}$

$$\widehat{y}_{M,low} = \begin{cases} \overline{\widehat{y}}_{T,low} + 3s_{low} & \text{if } \widehat{y}_{M,low} > \overline{\widehat{y}}_{T,low} + 3s_{low} \\ \overline{\widehat{y}}_{T,low} - 3s_{low} & \text{if } \widehat{y}_{M,low} < \overline{\widehat{y}}_{T,low} - 3s_{low} \\ \widehat{y}_{M,low} & \text{else} \end{cases} \tag{8.2.1}$$

in order to avoid overcompensation. The corrected quality prediction $\widehat{y}_C$ for $\widehat{y}_M$ is then given as

$$\widehat{y}_C = \widehat{y}_M - 0.75(\widehat{y}_{M,low} - \overline{\widehat{y}}_{T,low}), \tag{8.2.2}$$

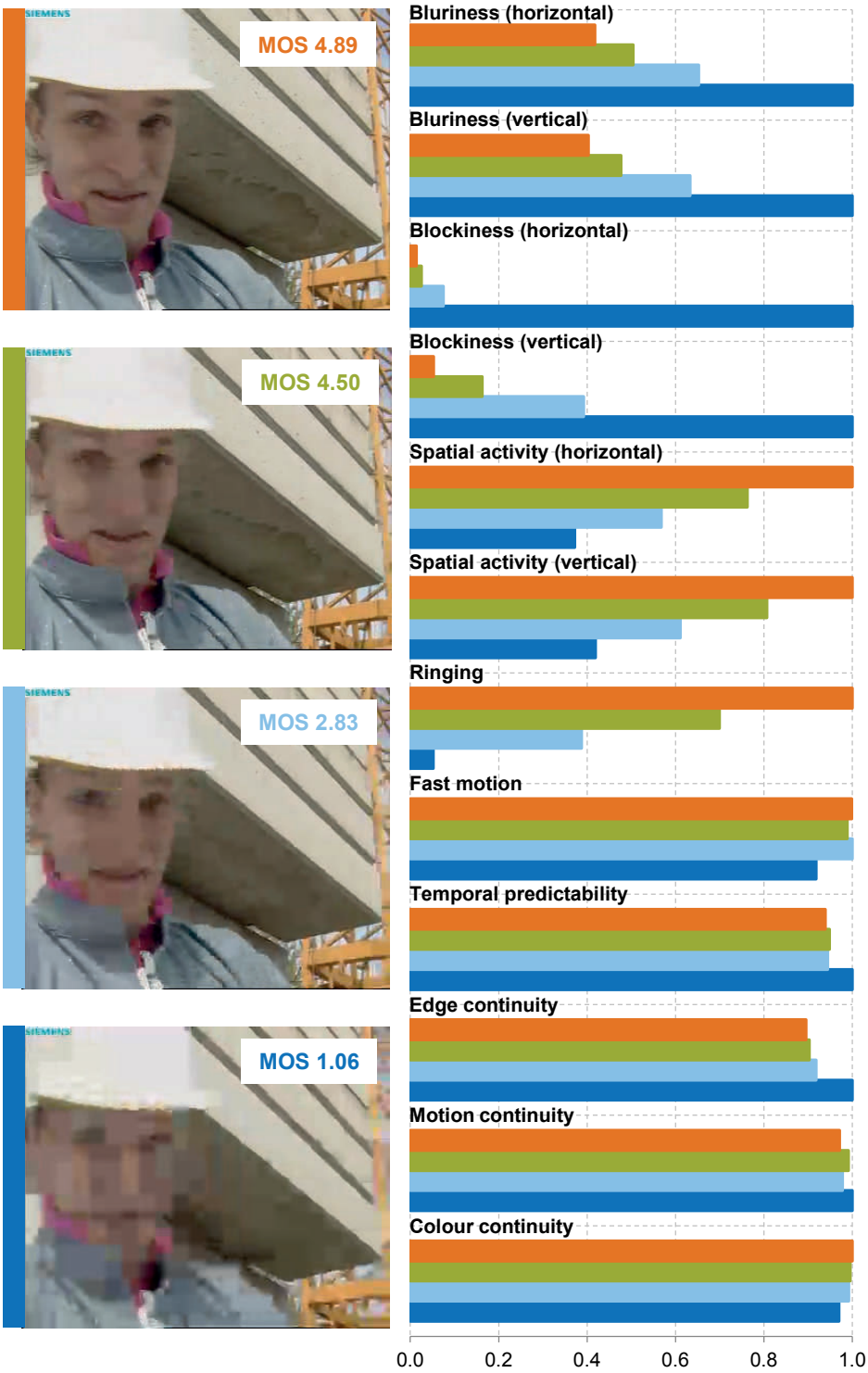

Figure 8.7.: Example of the variation within selected pixel-based features across different quality levels: *Foreman* at different visual quality from *bad* with a MOS of 1.06 to *excellent* with a MOS of 4.89. For comparability all features have been normalised to their maximum value

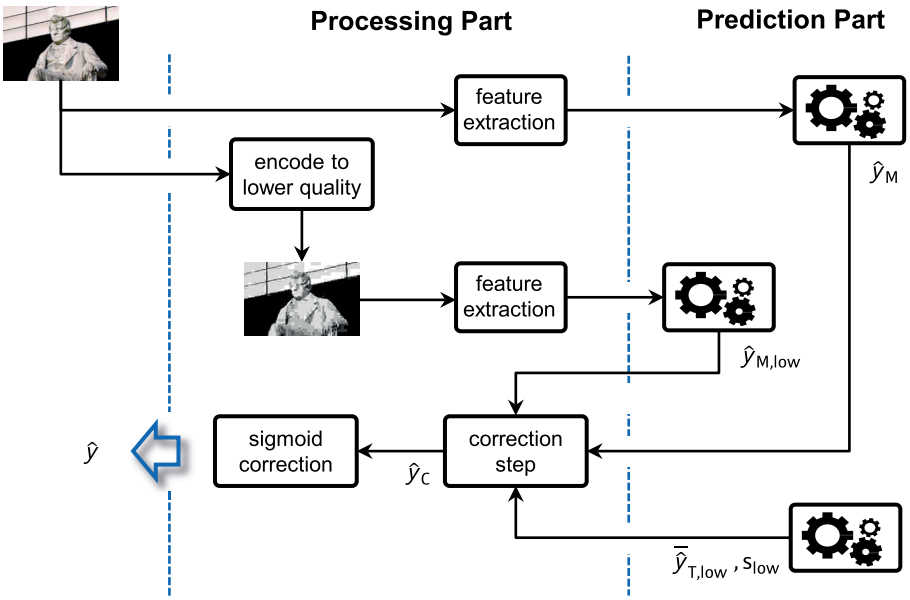

Figure 8.8.: Post processing for pixel-based example metric

where the factor 0.75 puts less weight on the correction step and more on the original quality prediction $\hat{y}_M$. As can be seen from (8.2.2), the correction term consists of the deviation of the quality prediction $\hat{y}_{M,low}$ of the video sequence's low quality version from the average quality $\bar{\hat{y}}_{T,low}$ of the low quality versions of the calibration videos. Hence we correct the quality prediction $\hat{y}_M$ with a measure of the current video sequence's sensitivity to coding artefacts compared to the average sensitivity of the video sequences used in building the prediction model and thus taking in to consideration the properties of the video under test. Note, that $\bar{\hat{y}}_{T,low}$ and s_{low} are fixed parts of the correction step that are determined during the model building and not fitted to the current video.

Finally, the same sigmoid correction as for the H.264/AVC design template in Section 8.1.3 is used

$$\hat{y} = \frac{a}{\left(1 + e^{-\frac{(\hat{y}_C - b)}{c}}\right)}, \tag{8.2.3}$$

resulting in the final quality prediction $\hat{y}$.

9. Performance Comparison

In this chapter, I compare the different data analysis methods discussed in this thesis by using them in the design of the example video quality metrics presented in the previous chapter.

Following an introduction of the performance metrics used to determine the individual video quality metric's performance, I outline the selection process of the data sets used to built the example metrics and their properties. As the data used for training and validation plays an important role in the data analysis approach, great care has to be taken that the basis of the models, the subjective quality ratings and the corresponding video sequences, are selected appropriately. Finally, the performance of the data analysis methods and their corresponding metric is compared in two parts: in the first part, the different data analysis methods are compared to each other, whereas in the second part the best data analysis method and corresponding metric is compared to selected state-of-the-art video quality metrics, before concluding with a short summary.

9.1. Performance metrics

This section briefly describes the performance metrics used in the performance comparison of the different data analysis and for a more detailed discussion I refer to Appendix B.3.

The aim of all these metrics is to provide a single, characteristic value, describing the prediction performance of a video quality metric with respect to the ground truth in the form of the subjective visual quality, described by the MOS. Or, in other words, how close the predicted visual quality $\widehat{\mathbf{y}}$ is to the true visual quality $\mathbf{y}$ for all sequences in a particular set of sequences.

9.1.1. Metrics

The used metrics are: the *Pearson correlation*, the *Spearman rank order correlation*, the *Kendall rank correlation*, the *root-mean-square error (RMSE)* and its derivative the *epsilon-insensitive RMSE* and lastly the *outlier ratio*.

Pearson correlation The correlation coefficient most commonly used in the research on video quality metrics is the *Pearson product-moment correlation coefficient* or *Pearson's r* [240, 241], often just called *Pearson correlation* and denoted r_p. It provides a measure of how strong the linear relationship between two variables, in our case $\widehat{\mathbf{y}}$ and $\mathbf{y}$, is.

Spearman rank order correlation Another correlation coefficient frequently used in the research on video quality metrics is the *Spearman rank order correlation coefficent* or *Spearman's* ρ [304, 305], often just called *Spearman rank correlation*. It is similar to the Pearson correlation, but instead of the variables themselves uses their corresponding rank. It provides a measure of the strength of the monotonicity between the two variables and is denoted as r_s.

Kendall rank order correlation Similar to the Spearman rank correlation, the *Kendall rank correlation* or *Kendall's* τ [149, 150], uses the rank of variables to determine a measure of their relationship and is denoted as r_k. But unlike the Spearman rank correlation that can be considered an extension of the Pearson correlation to ranks, the Kendall rank order correlation follows a different approach. It can be interpreted as an estimation of the probability that $\widehat{\mathbf{y}}$ is correctly ordered: if it has the same order as $\mathbf{y}$, r_k equals 1 but with increasing mismatch in the ranking between $\widehat{\mathbf{y}}$ and $\mathbf{y}$, the more r_k approaches 0.

Although the interpretation of the Kendall rank correlation as the probability of the correct order of a quality prediction is very attractive, Kendall's τ has so far only been used rarely in contributions to video quality metrics or visual quality estimation in general e.g. in [179, 258, 267].

Root-mean-square error (RMSE) The *root-mean-square error (RMSE)* is another commonly used performance metric in the field of video quality metrics. It provides a measure of the absolute error between the visual quality $\mathbf{y}$ and its prediction $\widehat{\mathbf{y}}$.

Epsilon-insensitive RMSE$_E$ The *epsilon-insensitive RMSE* or *RMSE$_E$* is an extension of the RMSE recently proposed by the Video Quality Experts Group (VQEG) [330]. Usually, the visual quality $\mathbf{y}$ is represented by the MOS of the corresponding sequences. But as already discussed in Section 2.5.2, the MOS itself is only an average of all votes and therefore does not represent the variance of votes. The idea is to take the confidence interval and therefore the uncertainty of the MOS into account when determining the prediction error with the RMSE.

Outlier ratio The last performance metric used in this thesis, is the *outlier ratio* or *OR*. This very simple performance metric represents the number of quality predictions $\widehat{y}$ that fall outside the 95% confidence interval of the corresponding subjective visual quality y as represented by the MOS.

9.1.2. Data fitting

In literature on video quality metrics, it is often suggested to fit the visual quality predicted by the metric, $\widehat{\mathbf{y}}$, to the visual quality gained in subjective testing, $\mathbf{y}$, with a non-linear function before applying the performance metric. Commonly used non-linear functions are monotonic logistic functions, e.g. in [284, 328, 329], or cubic polynomial functions, e.g. in [330, 344]. One argument in favour of this approach is related to the often used correlation measurements, e.g. the Pearson correlation. As these correlations are a measure of linear relationship, either between the values themselves or their rank, the relationship between the variables should be indeed linear. Hence, if the metric's prediction results exhibit a non-linear nature, an appropriate correction can map the results into a linear representation.

An additional argument often provided in favour of applying data fitting is best described by a quote from the Video Quality Experts Group (VQEG) [330, p.33]:

> Subjective rating data often are compressed at the ends of the rating scales. It is not reasonable for objective models of video quality to mimic this weakness of subjective data. Therefore, a non-linear mapping step was applied before computing any of the performance metrics

Yet, this so-called *weakness*, represents reality. Although the results of subjective testing often exhibit a non-linear shape in particular due to the contraction bias as discussed in Section 2.5.2, these results are the only data we have describing the visual quality as perceived by human observers. While it may very well be that the real visual quality is different from this recorded quality, it is not possible to quantify this influence. Hence, if a metric is expected to be an alternative to subjective testing, its prediction performance should not be judged based on a subsequent data fitting, but on how well it is able to predict the visual quality directly.

Additionally, there is a more general problem with the proposed fitting process: the fitting is only possible if the real visual quality is known from subjective testing. Obviously, such subjective data is not available in real-life applications, as the motivation to use video quality metrics is exactly to avoid subjective testing. Hence, the correlation coefficients gained after such a fitting are not necessarily representative of the real life performance of a video quality metric and tend to be overly optimistic as was already discussed in [144]. Consequently, no fitting of the video quality metrics' predictions will be performed in this thesis, but the performance metrics will be applied directly to the video quality metrics' quality.

9.2. Data sets

An alternative to conducting a subjective testing campaign is the use of existing and publicly available data sets. In this section, I define selection criteria for data sets to be used in the performance comparison and then evaluate the currently publicly available data sets with respect to these criteria, leading to the selection and brief review of the data sets used in this thesis.

9.2.1. Evaluation of publicly available data sets

In general, the main requirement on these data sets is that both the (distorted) video sequences and the corresponding MOS scores are available, as these two components are essential for the data analysis approach. Additionally, the data sets used in this thesis should also fulfil the following criteria:

1. Publicly available
2. Covering a sufficient range of content and quality levels
3. Individual subject's votes available
4. Progressive material
5. Only coding artefacts, no artefacts caused by transmission and/or packet errors
6. Include H.264/AVC Annex B bitstreams

The first point addresses the reproducibility of the results of this thesis, but also makes it possible to compare the performance with contributions in the state of the art using the same data sets. Note, that this point obviously also excludes all data sets not publicly available from e.g. MPEG standardisation or contract research for industrial partners. Secondly, the data sets should cover an adequate content and quality range representative of real-life applications, as this enables a realistic performance assessment of the different methods and metrics proposed in this thesis. Also data sets should be preferred that provide the individual subject's votes and not only the overall MOS scores, as this allows the use of additional performance metrics, in particular the epsilon-insensitive RMSE. The last three points address limitations of the H.264/AVC bitstream-based video quality metric that is used as an example in this thesis. Even tough this would not necessarily be an issue with the pixel-based video quality metric, the same data sets should be used for both example metrics in order to provide a common basis for the performance comparison. Hence, the focus is on typical artefacts introduced by encoding schemes e.g. blocking, blurring or ringing artefacts.

Considering these criteria, I briefly review the publicly available video quality data sets with respect to their suitability for the performance comparison in this thesis:

Video Quality Experts Group (VQEG) The Video Quality Experts Group (VQEG) provides two data sets: the older *VQEG FR-TV Phase I* data set [328] providing MPEG-2 compressed interlaced SDTV video sequences and the more recent *VQEG HDTV* data set [330] that provides both MPEG-2 and H.264/AVC compressed HDTV video sequences in different interlaced and progressive HDTV formats. The *VQEG FR-TV Phase I* needs to be excluded in this thesis due its lack of progressive material and only MPEG-2 compression according to criteria four and six, whereas the *VQEG HDTV* data set in principle fulfils all criteria, but the H.264/AVC bitstreams are not yet publicly available and hence this data set is also excluded.

Polytechnic Institute of New York University The video laboratory at the Polytechnic Institute of New York University presents two publicly available data sets in [231, 233–235] and [60, 184–186], with H.264/AVC Annex G (SVC) and H.264/AVC compressed video sequences with different packet loss rates, respectively. Both data sets include sequences in progressive CIF and QCIF format with different frame rates. Unfortunately, no bitstreams are provided and thus criterion six is not fulfilled.

IRCCyN/IVC The Institut de Recherche en Communications et Cybernétique de Nantes' (IRCCyN) Image and Video Communications (IVC) group provides numerous publicly available data sets. None, however, are suitable for the use in this thesis according to our criteria: the *IRCCyN/IVC SVC4QoE Replace Slice* [255], *IRCCyN/IVC SVC4QoE QP0 QP1* [257], *IRCCyN/IVC Temporal Switch* [254] and *IRCCyN/IVC H.264 AVC vs SVC VGA* [253] data sets provide video sequences in progressive QVGA and VGA format, but are on the one hand compressed with H.264/AVC Annex G (SVC), on the other hand no bitstreams are available, thus leading to their exclusion due to the sixth criterion. The *IRCCyN/IVC 1080i* [239] data set with video sequences in the 1080i HDTV format and the *IRCCyN/IVC H.264 HD vs Upscaling and Interlacing* [256] data set with video sequences in different SDTV and HDTV formats are both encoded with H.264/AVC, but are either at least partly in interlaced format or include transmission errors. Moreover, no bitstreams are available. Hence, we also have to reject these two data sets according to criteria four, five and six. Lastly, the *IRCCyN/IVC RoI* [18] and *IRCCyN/IVC Eyetracker SD 2009_12* [54] provide H.264/AVC encoded SDTV video sequences including the corresponding bitstreams, but as they contain interlaced material, we also have to reject these two data sets according to criterion number four.

EPFL & PoliMi The *EPFL-PoliMi video quality assessment* [295, 296] data set provided by a cooperation of the Ecole Polytechnique Federale de Lausanne (EPFL) and the Politecnico di Milano (PoliMI) contains H.264/AVC compressed video sequences and corresponding bitstreams in progressive CIF and 4CIF format. Only a small subset, however, is without packet loss, leading to only one usable data point for each video sequence. Additionally, EPFL provides the *MMSPG Scalable Video* [175, 176] data set containing

H.264/AVC Annex G (SVC) compressed video sequences in progressive 720p HDTV format and corresponding bitstreams. Although the H.264/AVC compliant base layer of the SVC bitstreams could be utilized, this would represent only a small subset of the data set. Hence, both data sets need to be excluded according to the fourth criterion.

University of Plymouth The *Audiovisual Quality Database for Mobile Multimedia Applications* [77] data set provided by the University of Plymouth contains H.263 compressed video sequences in progressive QCIF format, but all videos also include transmission errors caused by packet loss and the bitstreams are both unavailable and conform to a completely different coding technology. Therefore, we also have to reject this data set according to criteria five and six.

Data sets used in the comparison After considering the publicly available data sets and rejecting the data sets mentioned above, the following four data sets that fulfil the criteria well enough were selected and will therefore be used in the performance comparison: *TUM1080p50*, *TUM1080p25*, *LIVE Video Quality* and *IT-IST*. These four data sets represent a wide range of different content at different resolutions and frame rates, from IPTV and SDTV up to current and future HDTV formats.

Each data set will be described in detail in the following sections and a short overview over all data sets is also provided in Table 9.1 on the next page.

Winkler [352] provides a comprehensive in-depth analysis of many of the data sets discussed in this section by using multiple objective criteria, capturing content characteristics and providing a statistical analysis of the subjective ratings. For a comprehensive overview of publicly available data sets, I refer to the list maintained by Qualinet in [65, 66].

9.2.2. TUM1080p50

The *TUM1080p50* data set was generated during the work on this thesis in the ITU-R BT.500 [109] compliant video quality evaluation laboratory at the Institute for Data Processing at Technische Universität München and is representative of future HDTV formats. It contains five different H.264/AVC encoded, 10 s long video sequences from the well known SVT Multiformat Test Set [79] with a spatial resoultion of 1920×1080 pixel at 50 fps, covering a large range of content and coding difficulty. The used sequences are: *CrowdRun*, *TreeTilt*, *PrincessRun*, *DanceKiss* and *FlagShoot*. Key frames of the sequences are shown in Fig. 9.1 on page 154. All sequences were encoded with a high profile using the H.264/AVC JM reference software [292], version 17.1, at four bitrates, each depending on the coding difficulty of the individual sequences. This results in a bitrate range from 2 Mbit/s to 40 Mbit/s, representing real life high definition applications from the lower end to the upper end on the bitrate scale.

For subjective testing, the SSMM methodolgy as described in Section 2.5.2 with a 11-point discrete quality scale from 0, worst quality, to 10, best quality, was used. Using

Table 9.1.: Overview of data sets and video sequences used in the performance comparison. More details are provided in Appendix B.4

Data set	Format	fps	Data points	Video sequences	MOS range[a]
TUM1080p50	1080p	50	20	CrowdRun	1.9–8.0
				TreeTilt	3.0–8.8
				PrincessRun	1.9–7.0
				DanceKiss	2.9–8.0
				FlagShoot	2.5–7.7
TUM1080p25	1080p	25	32	CrowdRun	2.6–9.3
				ParkJoy	4.3–9.3
				InToTree	6.9–9.6
				OldTownCross	1.9–9.6
LIVE Video Quality	768×432	25	24	PedestrianArea	41–69
				RiverBed	39–64
				RushHour	38–63
				Sunflower	33–57
				Station	41–56
				Tractor	37–64
IT-IST Video Quality	CIF	25	48	Australia	2.2–4.9
		30		City	3.3–4.9
				Coastguard	1.8–4.7
				Container	3.7–4.9
				Crew	1.3–4.9
				Football	1.7–3.8
				Foreman	1.1–4.9
				Mobile	1.3–4.7
				Silent	1.6–5.0
				Stephan	1.0–4.9
				Table	1.2–4.9
				Tempete	2.6–4.9

[a] Discrete 11-point scale from 0–10, worst to best, for *TUM1080p50*/*TUM1080p25* data set
Continuous DMOS scale from 0–100, worst to best, for *LIVE Video Quality* data set
Discrete 5-point scale from 1–5, worst to best, for *IT-IST* data set

Figure 9.1.: Video sequences in the TUM1080p50 data set from top to bottom: *CrowdRun*, *TreeTilt*, *PrincessRun*, *DanceKiss* and *FlagShoot*

Figure 9.2.: Video sequences in the TUM1080p25 data set from top to bottom: *CrowdRun*, *ParkJoy*, *InToTree* and *OldTownCross*

the selected bitrates, this results in a quality range from *not acceptable* to *very good*, corresponding to a MOS between 1.9 and 8.9. The test was conducted with two different displays, a consumer LCD display and a reference LCD display, and a projector, resulting in overall three different subsets. In this thesis, I only use the subset of the results produced with the reference display. Thus we have in total 20 data points, each representing a combination of coding condition and content for the TUM1080p50 data set. The data set is available at [317] and for more information I refer to [136, 266].

9.2.3. TUM1080p25

Another result of the work on this thesis is the *TUM1080p25* data set. Like the *TUM1080p50* data set it originates in a subjective testing campaign conducted in the video quality evaluation laboratory at the Institute for Data Processing at Technische Universität München. It also uses four 10 s long video sequences from the SVT Multiformat Test Set [79] with a spatial resolution of 1920 $\times$ 1080 pixel, but contrary to the TUM1080p50 only with 25 fps and is thus representative of current HDTV formats. These lower frame rate versions of the video sequences were generated by dropping every even frame i.e. every second frame of the original 50 fps video material. The used sequences, *CrowdRun*, *ParkJoy*, *InToTree*, and *OldTownCross* represent a adequate range of content and coding difficulties. Key frames of the sequences are shown in Fig. 9.2 on the preceding page.

The video sequences were encoded with H.264/AVC at 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) main profile encoder, representative of standard devices, and the second setting, a high complexity (HC) high profile setting, aims at achieving the maximum possible quality with this coding technology, representing sophisticated broadcasting encoders. Similar to the TUM1080p50 data set, the H.264/AVC JM reference software [292], version 12.4, was used to encode the video sequences at for different bit rates from 5.4 Mbit/s to 30 Mbit/s, depending on the coding difficulty of the individual sequences. Additionally to H.264/AVC, the video sequences were also encoded with Dirac, an alternative, wavelet based video codec proposed by the British Broadcasting Cooperation (BBC) in [16, 17] at the same bitrates. For subjective testing, the DSUR methodology as described in Section 2.5.2 with a 11-point discrete quality scale from 0, worst quality, to 10, best quality, was used. Overall, the selected bitrates result in a quality range from *not acceptable* to *nearly perfect*, corresponding to a MOS between 1.9 and 9.6.

All in all, the set consists in total of 48 data points, each representing a combination of coding condition, content and different encoder setting or coding technology. In the end, however, only the data points relating to the H.264/AVC encoded video sequences can be used and thus only a subset of 32 data points will be used in the performance comparison in this thesis. The data set is available at [317] and for more information I refer to [136, 139].

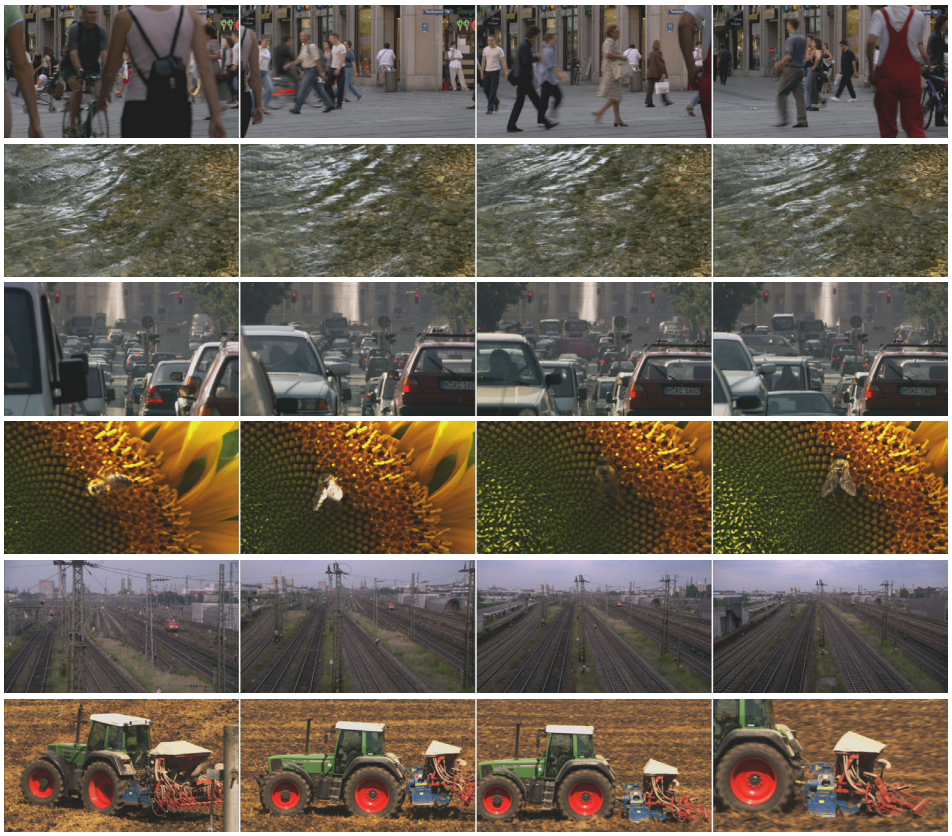

Figure 9.3.: Video sequences in the LIVE Video Quality data set from top to bottom: *PedestrianArea*, *RiverBed*, *RushHour*, *Sunflower*, *Station* and *Tractor*

9.2.4. LIVE Video Quality

The next data set that will be used in this thesis is the *LIVE Video Quality* data set that originates at the Laboratory for Image and Video Engineering (LIVE) at The University of Texas at Austin. It consists of ten different video sequences with an original spatial resolution of 1920 $\times$ 1080 pixel and 1280 $\times$ 720 pixel for the 25 fps and 50 fps material, respectively. They were resized to a spatial resolution of 768 $\times$ 432 pixel, preserving the 16:9 aspect ratio. This resolution is close to common SDTV formats like 576i or 480i and therefore this dataset can be considered to be representative of SDTV formats. From the ten sequences, three sequences, *ParkRun*, *Shields* and *Mobile&Calendar*, have a frame rate of 50 fps and seven sequences, *BlueSky*, *PedestrianArea*, *RiverBed*, *RushHour*, *Sunflower*, *Station* and *Tractor*, have a frame rate of 25 fps. All sequences except BlueSky are 10 s long. In the end, only the larger subset with a frame rate of 25 fps will be used in this thesis and additionally BlueSky from the 25 fps subset was excluded as this sequence

is significantly shorter with only 8.68 s. Key frames of the used sequences are shown in Fig. 9.3 on the facing page.

The LIVE Video Quality data set considers four different distortion classes: MPEG-2 encoding, H.264/AVC encoding, H.264/AVC encoding with packet loss in IP networks and H.264/AVC encoding with packet loss in wireless networks. The H.264/AVC encoding class consists of the video sequences encoded at four bitrates from 200 kBit/s upto 5 MBit/s using the H.264/AVC JM reference software [292], version 12.3. For subjective testing, a single stimulus method with hidden reference removal, similar to [330], was used. After normalisation in the post processing of the results this resulted in a Difference Mean Opinion Score (DMOS) on continuous quality scale between 0, worst quality, to 100, best quality. Overall, the selected bitrates result in a quality range between 33 and 69.

Of these four classes, only the subset with H.264/AVC encoding and the six sequences of the 25 fps subset with 10 s length will be used, resulting in 24 data points. Unlike the other data sets, only the overall DMOS' are provided and not the individual votes of the test subjects. Although the data set should therefore be excluded, it is still included, as it provides video sequences with spatial and temporal resolution close to common SDTV resolutions. It thus helps to cover the complete range from IPTV up to HDTV. The data set is available at [164] and for more information I refer to [284, 285].

9.2.5. IT-IST

The last and also the largest data set that will be used in this thesis is the *IT-IST* data set created at the Instituto de Telecomunicações at the Instituto Superior Técnico in Lisbon. It consists of twelve different, H.264/AVC encoded video sequences with a spatial resolution of 352 $\times$ 288 pixel at 25 fps and 30 fps. It is therefore a good representative of IPTV formats in CIF resolution. From the twelve sequences, one sequence, *Australia*, has a frame rate for 25 fps, and all other sequences,*City*, *Coastguard*, *Container*, *Crew*, *Football*, *Foreman*, *Mobile*, *Silent*, *Stephan*, *Table* and *Tempete*, have a frame rate of 30 fps. Even tough the sequence Australia has a different frame rate at 25 fps and should therefore be excluded, it was used in the design of the bitstream-based metric, as the coding structure of its H.264/AVC bitstream is identical to the other sequences with 30 fps. It was, however, excluded for the pixel-based metric due to the mismatch between playback rate and intra-frame rate, as the pixel-based metric considers features at the original playback rate and the results from Australia would therefore not align with the other sequences, impacting the model building process of the metric negatively. Key frames of the used sequences are shown in Fig. 9.4 on the next page and Fig. 9.5 on page 159.

All sequences were encoded with the H.264/AVC JM reference software [292], version 12.4, using a main profile. The sequences *Australia*, *City*, *Container*, *Crew*, *Foreman*, *Silent*,*Table* and *Tempete* were encoded at four different bitrates, whereas the sequences *Coastguard*, *Football*, *Mobile* and *Stephan* were encoded at six different bitrates. This results in a bitrate range from 64 kBit/s to 2 Mbit/s, covering a large quality range. Because of limitations in the prediction model building process that requires the same number of

Figure 9.4.: Video sequences in the IT-IST data set from top to bottom: *Australia*, *City*, *Coastguard*, *Container* and *Crew*

data points for all sequences, only four bitrates were used for each sequence and therefore two bitrates for *Coastguard*, *Football*, *Mobile* and *Stephan* each were excluded.

For subjective testing, the DCR methodology according to ITU-T P.910 as described in Section 2.5.2 with a 5-point discrete impairment scale from 1, very annoying, to 5, imperceptible, was used. Overall, the selected bitrates result in a quality range from *not acceptable* to *perfect*, corresponding to a MOS between 1 and 5. The sequences and corresponding bitrates result in a total of 48 data points, each representing a combination of coding condition and content. The data set is available at [95] and for more information I refer to [23–25].

Figure 9.5.: Video sequences in the IT-IST data set from top to bottom (continued): *Football*, *Foreman*, *MobileSilent*, *Stephan*, *Table* and *Tempete*

9.3. Comparison of two-way and multi-way data analysis

In this section, I compare the prediction performance of the two-way and three-way data analysis methods introduced in Chapter 5 and Chapter 6 by building prediction models with the selected data sets for both example metrics. Hence prediction models are built with temporally pooled three-way data and *MLR*, *PCR* and *bilinear PLSR*, and with three-way data and *unfolding in combination with bilinear PLSR*, *2D-PCR* and *trilinear PLSR*. Although unfolding can be used with any two-way data analysis method discussed in this thesis, I only use the PLSR in combination with unfolding. This choice is motivated by the fact that the PLSR is the only two-way method that takes the visual quality into account during the model building.

Methodology Before starting the model building process, the subjective visual quality $\mathbf{y}$ represented by the MOSs in the data sets were rescaled to the range of $[0, 1]$ for better comparability of the results between the different data sets. Similarly, the features in the three-way feature array $\underline{\mathbf{X}}$ were autoscaled i.e. centred across the first mode and scaled within the second mode. For the two-way data analysis methods necessitating the temporal pooling of $\underline{\mathbf{X}}$ into a two-way feature array $\mathbf{X}$, the autoscaling was applied after the temporal pooling by averaging over all frames. No data fitting was performed for the visual quality predictions $\widehat{y}$.

Using leave-one-out cross validation, separate prediction models were then built and validated for each of the four datasets with the six different two-way and multi-way data analysis methods. Therefore for each data set with N sequences consisting of I different sources encoded at $J = N/I$ bitrate points, I-different models were built with $IJ - J$ sequences for each data analysis method. For the comparison between the different data analysis methods, the number of components R was chosen as the global minimum encountered in the PRESS plot for any of the methods, and this number of components was then also used for all other data analysis methods. Hence, we compare all two-way and three-way data analysis methods with models that use the same number for components.

In the data sets containing sequences with different length, the number of frames in the shortest sequence was used for all sequences, as the dimension of the third mode in the three-way feature array $\underline{\mathbf{X}}$ representing the frames is required to be the same for all sequences. By using the shortest sequence and thus the lowest number of frames, we are able to use an increased number of sequences in the cross validation. Hence for the IT-IST, TUM 1080p25 and TUM 1080p50 data sets, $T = 259$, $T = 248$ and $T = 491$ frames were used, respectively, and for the LIVE data set all $T = 250$ frames were used.

Implementation of the algorithms All data analysis methods were implemented in MATLAB and for the performance comparison the models were built using MATLAB R2012b [204]. For the MLR, the integrated MATLAB function for the Moore-Penrose pseudo-inverse, `pinv`, was used, and the PCR was implemented using the integrated

MATLAB function for the SVD, `svd`. The 2D-PCR as described in Algorithm 6.2 was also implemented using the the `svd` function. For the PLSR, the `plsregress` function of the MATLAB Statistics Toolbox was used that implements de Jong's SIMPLS algorithm as described in Algorithm 5.4. For the trilinear PLSR, the implementation in the N-way Toolbox for MATLAB by Andersson and Bro [5], version 3.20 was used.

The feature extraction for the two example no-reference metrics was done separately from the model building using dedicated programs: the H.264/AVC bitstream features were extracted using the modified H.264/AVC JM decoder by Klimpke et al. [153] and the pixel-based features were extracted using a modified version of the software originally developed by Oelbaum [226].

Presentation of the results For brevity, only the PRESS plots and scatter plots for the IT-IST data set are provided in this section, as it is the largest and content-wise most comprehensive data set. In the PRESS plots, the RMSEP is denoted simply as RMSE as the RMSEP is equivalent to the RMSE in the performance comparison. For the TUM1080p25, TUM1080p50 and LIVE data set, the PRESS and scatter plots are provided in Appendix B.5 and Appendix B.6 for the bitstream- and pixel-based example metric, respectively. Also the OR, RMSE and RMSE_E are expressed as the complement to their maximum value of 1 i.e. $1 - \text{OR}$, $1 - \text{RMSE}$ and $1 - \text{RMSE}_E$, respectively, for easier interpretability of the plots.

9.3.1. Bitstream-based metric example

The H.264/AVC bitstream-based no-reference example metric uses $M = 17$ features extracted from each frame of the video sequences in the data set, resulting in a three-way feature array $\underline{\mathbf{X}} \in \mathbb{R}^{N \times 17 \times T}$, where N and T are dependent on the used data set.

Component selection Using the PRESS plot shown in Fig. 9.6 on the next page, $R = 2$ components were chosen as the optimal number of components for the metrics built with the IT-IST data set. Although the RMSE for the combination of unfolding and bilinear PLSR is slightly better for $R = 3$ components, this difference is less than 5×10^{-3} compared to the earlier minimum for $R = 2$ components and $R = 2$ was therefore chosen in line with the principle of parsimony. Similarly, for the TUM1080p25, TUM1080p50 and LIVE data set, the optimal number of components was determined as $R = 5$, $R = 2$ and $R = 1$, respectively.

Prediction performance The evaluation of the prediction performance of the different data analysis methods with respect to the performance metrics is shown for all data sets in the radar charts in Fig 9.7 on page 163 and additionally for the IT-IST data set as a scatter plot in Fig. 9.8 on page 164. The results of this evaluation are also available in Table B.6 in the Appendix.

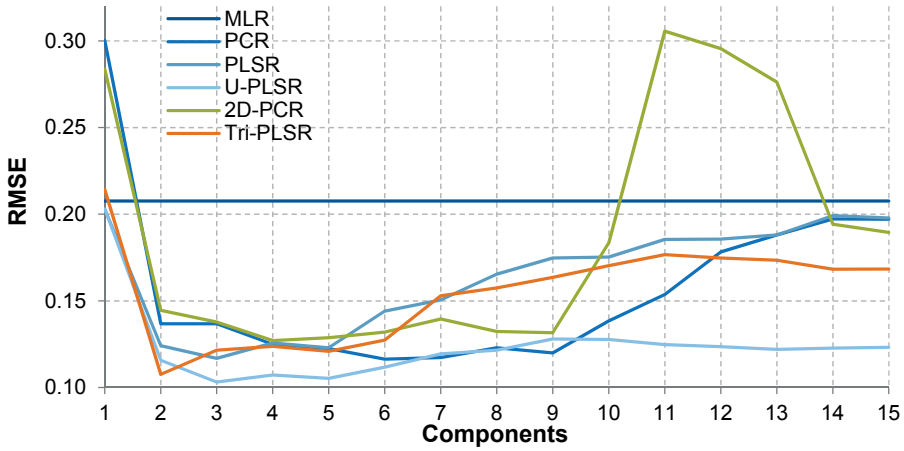

Figure 9.6.: PRESS plot for bitstream-based example metric and the IT-IST data set in order to determine the optimal number of R components for the prediction error as expressed by the RMSE for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR). For clarity only the first 15 components are shown

In the radar charts in Fig. 9.7, a *perfect* metric would be represented as regular hexagon with a radius of one. Thus the larger and more regular the hexagon spanned by a metric is, the better the overall prediction performance of a metric is and consequently if a metric envelops another metric, the enveloping metric outperforms the enclosed metric.

The trilinear PLSR outperforms all other two-way and three-way methods for the IT-IST, TUM1080p25 and TUM1080p50 data sets as shown in Fig. 9.7 and considering all data sets, the best prediction performance is achieved for the IT-IST data set with a Pearson and Spearman rank correlation of higher than 0.95. Additionally, we can observe that methods using three-way data without pooling in general perform better than those necessitating the temporal pooling of the features and depending on the data set, unfolding and bilinear PLSR or 2D-PCR rank second best with respect to most performance metrics. This is also confirmed by the scatter plot of the visual quality prediction $\widehat{y}$ versus the visual quality y for the IT-IST data set in Fig. 9.8, where the methods utilising the three-way data achieve a better fit to the desired linear relationship. The results not only confirm the inferiority of the MLR compared to component based methods that exploit the latent variables within the data, but also that for the two-way component methods PLSR outperforms the PCR. The difference in the prediction results of the trilinear PLSR and the MLR with respect to the Pearson correlation, Spearman rank correlation and RMSE are also statistical significant at the 0.05 level with a p-value of $p < 0.05$ for the IT-IST, TUM1080p25 and TUM1080p50 data sets. Unfortunately this can not be confirmed for differences in the prediction performance between the other methods and the results of the significance testing are provided in detail in Fig. B.9 in the Appendix.

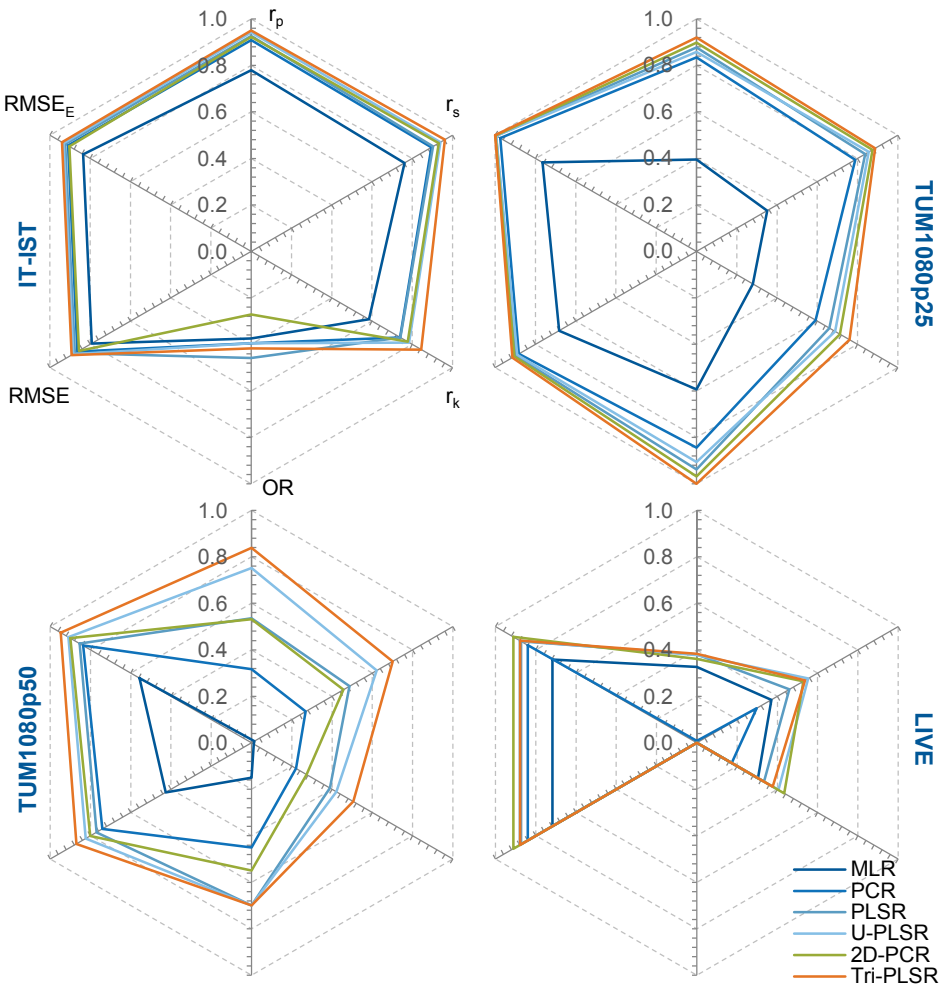

Figure 9.7.: Prediction performance for the bitstream-based example metric of MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR) with respect to the Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive RMSE_E. For clarity only the axes for the IT-IST data set are labelled

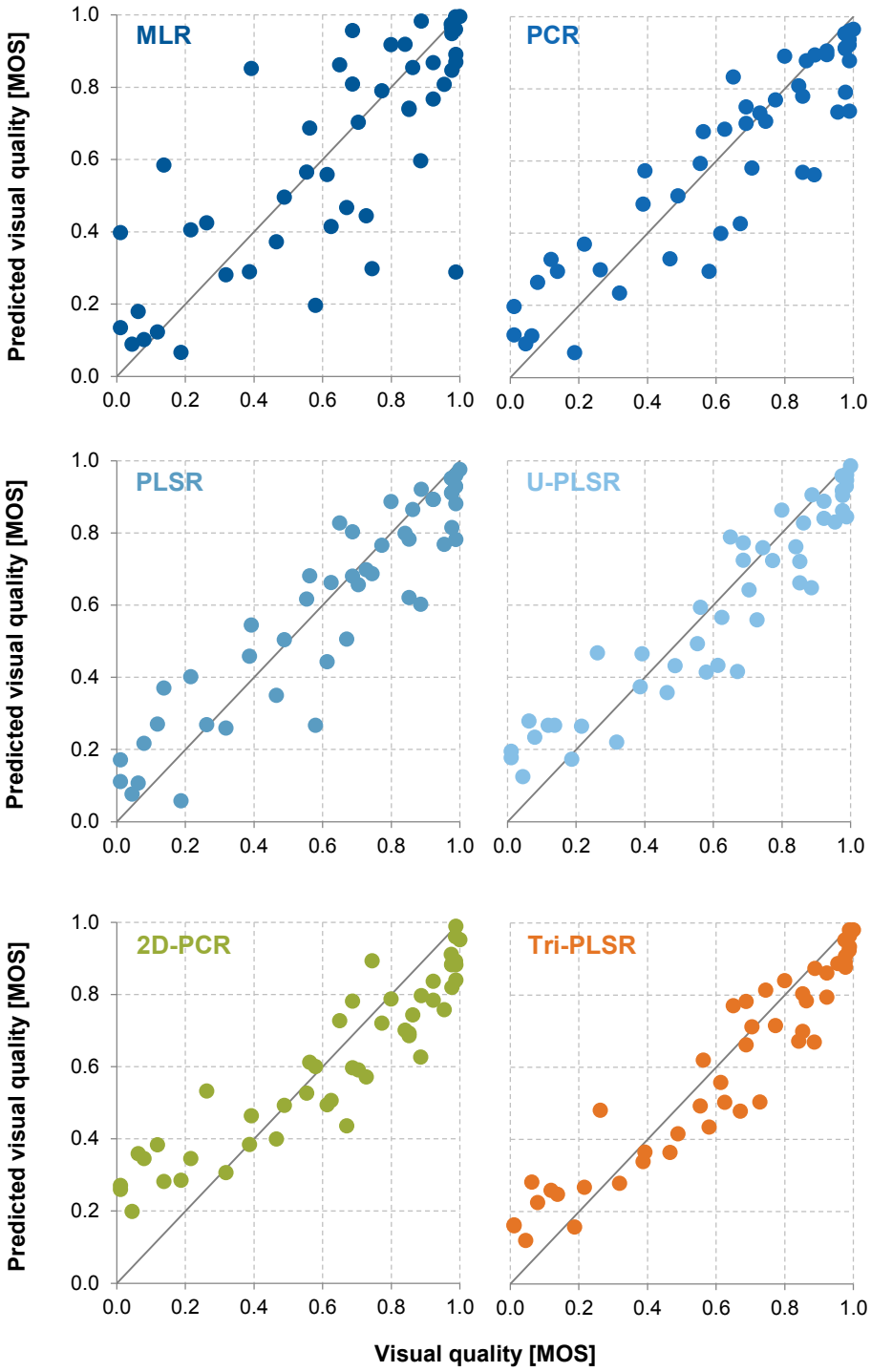

Figure 9.8.: Scatter plots of the prediction results of the bitstream-based example metric for the IT-IST data set, showing the visual quality y the against predicted visual quality $\hat{y}$ for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

In contrast to the IT-IST, TUM1080p25 and TUM1080p50 data sets, however, the prediction performance for the LIVE dataset is disappointing. Even tough for this data set also the methods utilising the three-way data are better than those using the pooled features, the overall performance except for the RMSE is not acceptable with a Pearson correlation and Kendall rank correlation below 0.4.

One possible explanation is that the LIVE data set due to its normalisation approach for the MOSs reduced the variation within **y** and thus methods exploiting the variation within **y** can not gain additional insight into the latent structure. This argument is supported by the fact that for the LIVE data set the prediction results for MLR and the PLSR-based methods are much closer than in the other data sets. In the analysis of the LIVE data set by Winkler [352], a lower median standard deviation in its MOSs compared to other video data sets was observed, further supporting this argument. Similarly, the correlation between PSNR and MOS for the LIVE data set was also significantly lower than for other data sets and the standard deviation of the PSNR smaller, both indicating that the physical measurable features exhibit low variance. Examining the extracted bitstream-features, the features' standard deviation corresponding to versions of the same content at different MOSs is very similar for all versions unlike for the other data sets. Thus also the variance within the features with respect to different quality levels is rather low, further limiting the extraction of latent variables based on the variance within the features. Note, that no confidence intervals are available for the LIVE data set and therefore for the LIVE data set all predictions are outliers and the RMSE is equal to the RMSE_E.

Another anomaly seems to be the rather poor performance of all data analysis methods for the IT-IST data set with respect to the outlier ratio, but this can be explained by the comparably smaller confidence intervals for this data set compared to the TUM1080p25 and TUM1080p50 data sets.

Influence of the data set composition One of the selection criteria for the data sets discussed in Section 9.2.1 is a sufficiently large range of content, allowing us to calibrate our model well enough for the unknown content in the validation. If our prediction model, however, encounters a content type or distortions in the validation it didn't encounter in the calibration, the prediction model will not be able provide an adequate quality prediction. For the bitstream-based example metric excluding the results of the LIVE data set, we can notice that the TUM1080p50 data set performs worse than the other two data sets, in particular compared to the TUM1080p25 data set, even tough both data sets share the same coding conditions and also similar content.

Taking a closer look at the content in the sequences of the TUM1080p50 data set, it becomes clear that the *FlagShoot* sequence is significantly different compared to the other sequences in the data set, as unlike all other sequences it contains a scene cut. This consequently leads to significant changes in the properties of the bitstream features before and after the cut e.g. with respect to the average motion vector length. But as a similar change does not occur in the other sequences, the prediction model built with the

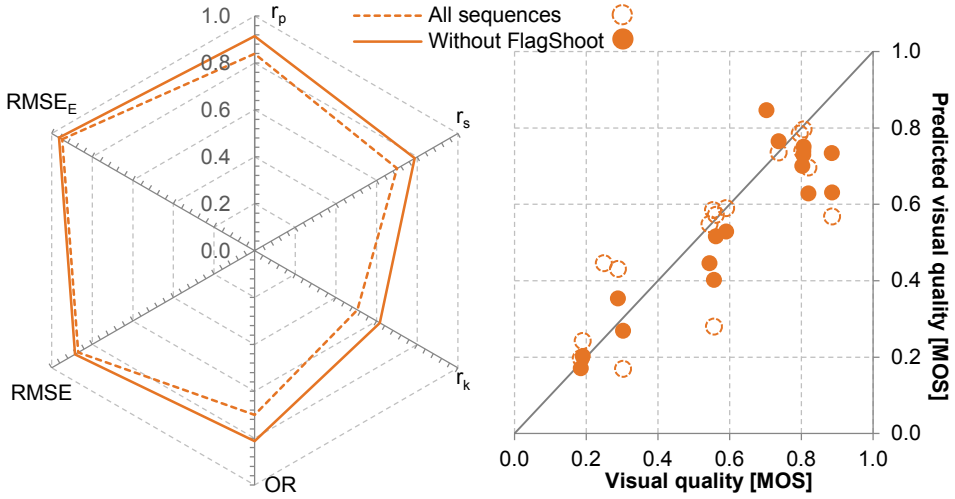

Figure 9.9.: Influence of the data set composition on the prediction performance: trilinear PLSR (Tri-PLSR) and the TUM1080p50 data set as an example. Prediction performance with all sequence and with the sequence *FlagShoot* excluded with respect to Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive $RMSE_E$ on the left, and corresponding scatter plot on the right

other sequences does not anticipate such an abrupt change within a sequence and thus the prediction performance suffers.

Similarly, a prediction model that includes *FlagShoot* in the calibration will provide a suboptimal prediction model for the other sequences in the cross validation. By excluding *FlagShoot*, the prediction performance will therefore increase as, for example, illustrated for the prediction model built with the trilinear PLSR in Fig. 9.9, where in particular the Pearson correlation increases from 0.84 to 0.91. Thus this underlines the importance of the data set composition with respect to the sufficiently large coverage of different content types that will be encountered in the validation or real-life application.

Importance of cross validation In the requirements on video quality metrics in Chapter 3, one of the requirements for compliance was the separation of calibration and validation sequences or a cross validation approach. The importance of this requirement can be demonstrated impressively on the example of the MLR and the TUM1080p25 data set. In the performance comparison for the TUM1080p25 data set, the MLR performed significantly worse than all other methods, but using the same sequences for calibration and validation and thus no cross validation, the prediction performance is nearly perfect as illustrated in Fig. 9.10 on the next page. Clearly, this is misleading, as in a real-life application scenario a video quality metric will rarely encounter the exactly same sequences it was calibrated on.

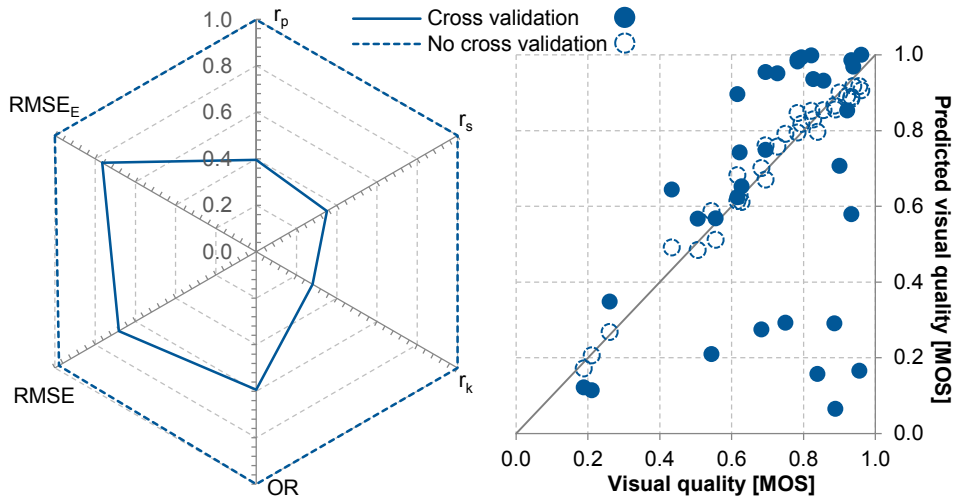

Figure 9.10.: Importance of cross validation: MLR and the TUM1080p25 data set as an example. Prediction performance of the MLR with and without cross validation with respect to Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive RMSE$_E$ on the left, and corresponding scatter plot on the right

9.3.2. Pixel-based metric example

The pixel-based no-reference example metric uses $M = 12$ features extracted from each frame of the video sequences in the data set and as some of the features require knowledge of the proceeding or succeeding frame, the first and last frame of the sequences are excluded, resulting in a three-way feature array $\underline{\mathbf{X}} \in \mathbb{R}^{N \times 12 \times T-2}$, where N and T are dependent on the used data set.

Component selection Using the PRESS plot shown in Fig. 9.11 on the following page, $R = 6$ components were chosen as the optimal number of components for the metrics built with the IT-IST data set. Note, that for $R = M$ Fig. 9.11 also shows the equivalence of MLR, PCR and PLSR as discussed in Section 5.4. Similarly, for the TUM1080p25, TUM1080p50 and LIVE data set, the optimal number of components was determined as $R = 1$, $R = 3$ and $R = 1$, respectively.

Prediction performance The evaluation of the prediction performance of the different data analysis methods with respect to the performance metrics is shown for all data sets in the radar charts in Fig 9.12 on page 169 and additionally for the IT-IST data set as a scatter plot in Fig. 9.13 on page 170. The results of this evaluation are also available in Table B.7 in the Appendix.

Unlike for the bitstream-based example metric, the trilinear PLSR does not outperform the other methods for all data sets, but only for the IT-IST and TUM1080p25 data sets. For

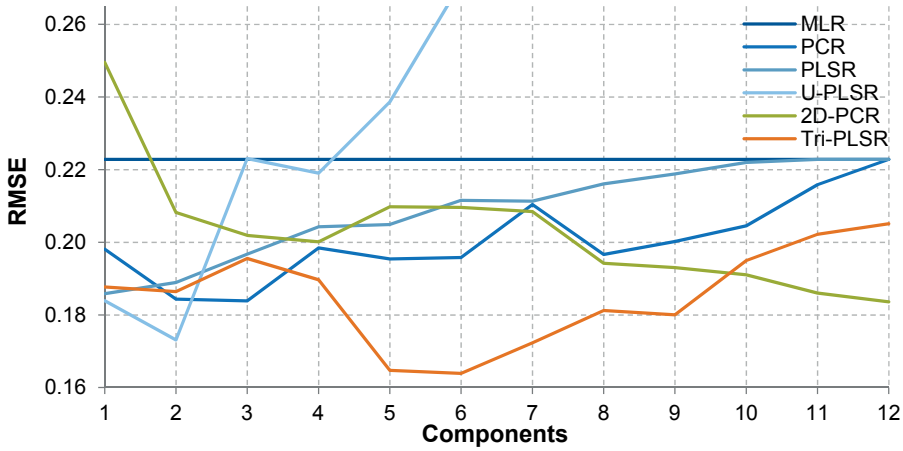

Figure 9.11.: PRESS plot for pixel-based example metric and the IT-IST data set in order to determine the optimal number of R components for the prediction error as expressed by the RMSE for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

the TUM1080p50 data set, the combination of unfolding and PLSR provides a significantly better prediction performance as seen in Fig. 9.12. Also the results depend more on the individual method and we can no longer confirm in general that methods using three-way data without pooling perform better than those necessitating the temporal pooling. Still, the best performing method in each data set uses three-way data without pooling and is either the trilinear PLSR or unfolding in combination with PLSR.

The overall prediction performance for all prediction sets is lower than for the bitstream-based example metric and the best prediction performance is achieved for the IT-IST data set with a Pearson correlation of 0.88. This result is in line with the evaluation of existing video quality metrics in Chapter 3, where all bitstream-based metrics exceeded a Pearson correlation of 0.9, but only a significantly smaller subset of pixel-based metrics was able to achieve this goal. The differences in the prediction results of the component based methods and the MLR with respect to the RMSE are also statistical significant at the 0.05 level with a p-value of $p < 0.05$ for the TUM1080p25 and TUM1080p50 data sets. Unfortunately this can not be confirmed for differences in the prediction performance between the other methods and the results of the significance testing are provided in detail in Fig. B.16 in the Appendix.

Similar to the bitstream-based example metric, the results for the LIVE data set are also significantly worse compared to the other data sets and compared to the bistream-based features, the example metric based on pixel-based features is even worse as the MLR achieves the best performance with respect to the correlation coefficients. Examining the pixel-based features in a similar way as before the bitstream-based features, the results indicate that the variation within the pixel-based features is even lower than for the bitstream-

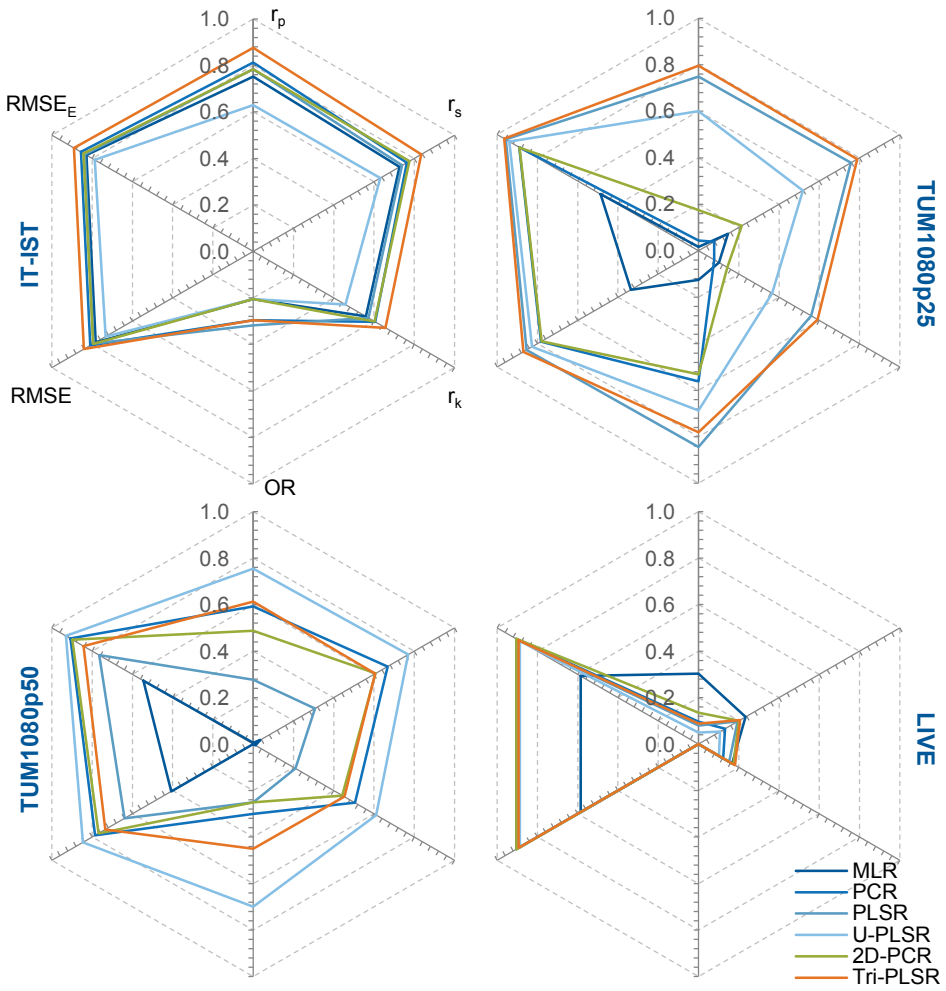

Figure 9.12.: Prediction performance for the pixel-based example metric of MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR) with respect to the Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive RMSE_E. For clarity only the axes for the IT-IST data set are labelled

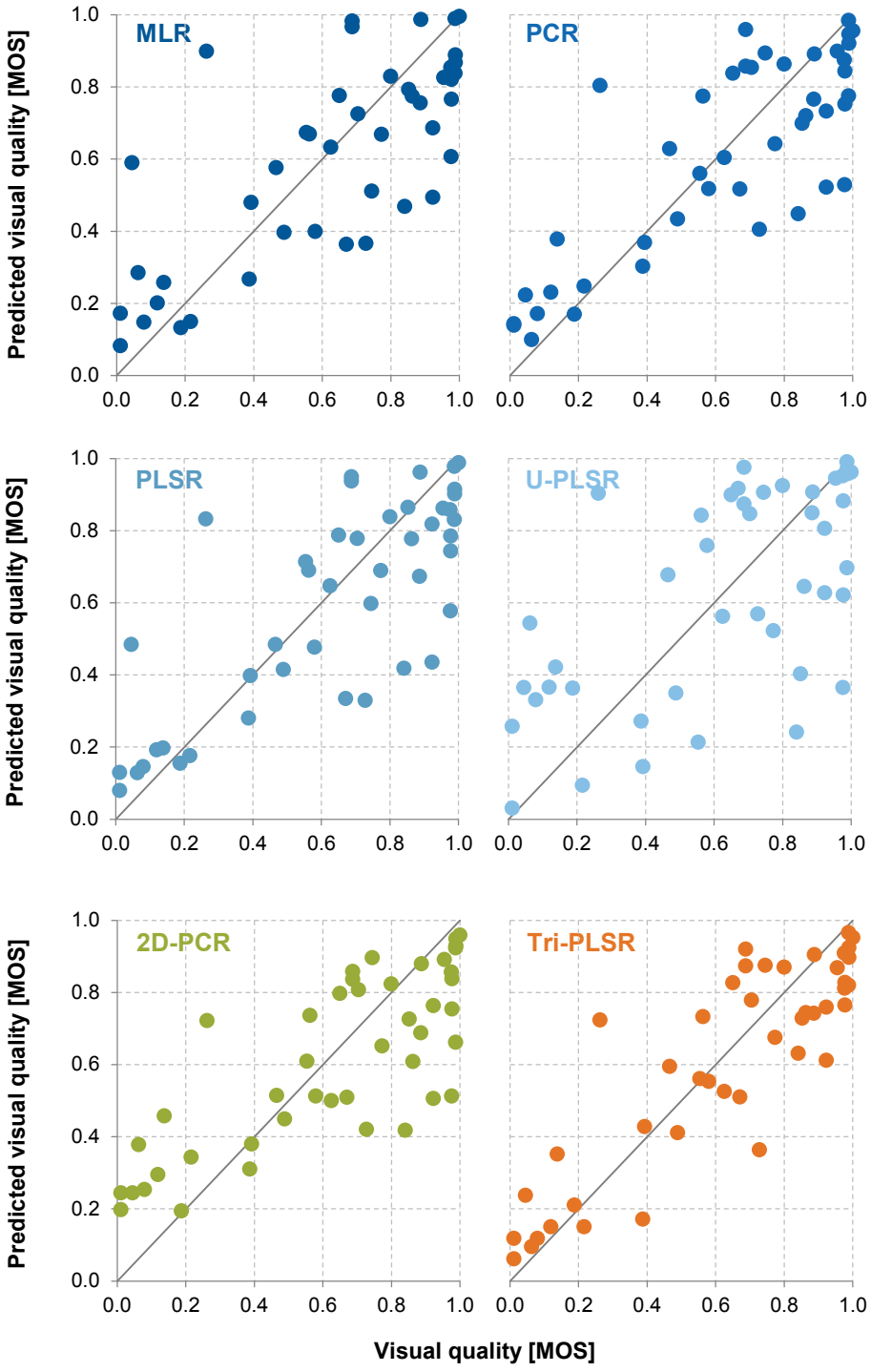

Figure 9.13.: Scatter plots of the prediction results of the pixel-based example metric for the IT-IST data set, showing the visual quality y the against predicted visual quality $\widehat{y}$ for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR).

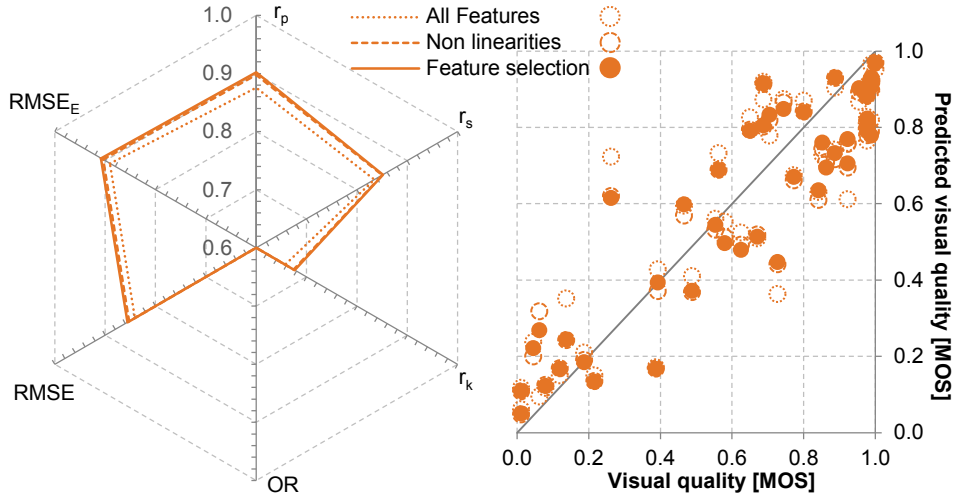

Figure 9.14.: Influence of non-linearities and feature selection on the prediction performance: trilinear PLSR and the IT-IST data set as an example. The prediction performance is shown for different models with all features, non-linearly pooled features and non-linearly pooled features in combination with only selected features. Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive RMSE$_E$ on the left, and corresponding scatter plot on the right

based features, explaining the even worse performance of the component-based methods for the pixel-based example metric compared to the bitstream-based example metric.

Non-linearities and feature selection Even tough the aim of the two example metrics in this thesis is only to enable us to compare the different two-way and three-way data analysis methods and not to provide a highly optimised video quality metric, it is illustrative to discuss two common strategies to optimise a data analysis based metric. The first optimisation approach takes into account non-linearities and the second approach focuses on selecting only features contributing to the prediction as discussed in Section 5.5 and Section 7.4, respectively.

Some pixel-based features are determined for both the horizontal and vertical dimension of a frame $\mathbf{S}_t$, in particular the blockiness and bluriness. Assuming that the direction of the blockiness or bluriness is less important than the magnitude and the overall blockiness or bluriness is dominated by the maximum of either the horizontal or vertical component, one of the two direction dependent values is redundant, and we can pool the horizontal and vertical blockiness with a non-linear maximum operator into one, direction independent blockiness or bluriness feature. Using the trilinear PLSR and the IT-IST data set as an example, the prediction performance can be increased for all performance metrics as illustrated in Fig 9.14. In particular, the Pearson correlation increased from 0.876 to 0.898, nearly fulfilling our requirement on prediction performance. This optimisation, however, re-

quires some knowledge about possible non-linear relationships between the features that is not always available.

Feature selection is the second optimisation strategy that can be applied to the prediction model. It aims at a more parsimonious prediction model by excluding features that do not contribute significantly to the prediction. Using either simple leave-one-out cross validation with respect to the features or one of the methods discussed in Section 7.4 for the feature selection, the feature *edge continuity* emerges as non-influential for the prediction model and is therefore excluded in the model building process. Although the additional gain of the prediction built without this feature is rather small compared to the consideration of non-linearities, the feature selection results in a further improvement of the prediction performance as illustrated in Fig. 9.14 on the preceding page. Together with the non-linear pooling we achieve a Pearson correlation of 0.902, allowing us to fulfil the requirement on prediction performance. In contrast to the consideration of non-linear effects, the feature selection can be performed without any further knowledge of features' properties, as only their influence on the prediction model is used in the selection process.

9.4. Comparison of example metrics to the state-of-the-art

Following the comparison of the prediction models built with the different data analysis methods and the two example metrics, I compare in this section the data analysis approach with the state-of-the-art video quality metrics. The data analysis approach is represented by the best performing combination of data analysis method and example metric, the *bitstream-based metric* built with *trilinear PLSR*. The video quality metrics representing the state-of-the-art are the *PSNR*, the *Structural SIMilarity index (SSIM)* [338], the *Information content Weighted MS-SSIM (IW-SSIM)* [340], the *Natural Image Quality Evaluator (NIQE)* [208] and the *MOtion-based Video Integrity Evaluation index (MOVIE)* [282].

Selection criteria for metrics The two main criteria for the selection of the video quality metrics are that the metrics should on the one hand represent the variety of the existing state-of-the-art well enough, but on the other hand implementations of the metrics should also be publicly available. The latter is not only to avoid implementing the metrics, but also owed to the fact that it is unfortunately often not possible to implement an algorithm just based on literature.

Although the aim of this comparison should be to compare the two example no-reference video quality metrics in this thesis with no-reference video quality metrics in the state-of-the-art, this goal is not feasible, as implementations of neither no-reference visual quality metrics in general nor video quality metrics in particular are widely available. Thus mostly image quality metrics are used in this comparison.

PNSR and the SSIM are included in the comparison to represent the currently two most commonly used full-reference image quality metrics. Even though PSNR should not be considered a visual quality metric, it is still often used as such in literature and therefore

included in the comparison. The IW-SSIM represents the multitude of existing SSIM variants and is also one of the best performing SSIM variants according to the comprehensive comparison in [340]. NIQE is one of the latest no-reference image quality metrics using a natural scene statistics approach to video quality metrics and additionally one of the few no-reference metrics with a publicly available implementation. Finally, the full-reference video quality metric MOVIE represents metrics with a moderate consideration of the temporal domain. For more details on the selected metrics, I refer to Section 3.3.2.

Implementation of the metrics Both the PNSR and SSIM were determined using the *Video Quality Measurement Tool (VQMT)* by Hanhart [81]. For the IW-SSIM, the reference implementation by Wang and Li [340] available at [335] was used, and for NIQE and MOVIE the reference implementations by Mittal et al. [208] and Seshadrinathan and Bovik [282], respectively, both available at [165], were used.

Methodology In order to ensure the comparability of the prediction performance to the data analysis based metrics, the number of frames used in the video sequences for the quality prediction was the same as in Section 9.3: for the IT-IST, TUM 1080p25 and TUM 1080p50 data sets this resulted in $T = 259$, $T = 248$ and $T = 491$ frames, respectively, and for the LIVE data set all $T = 250$ frames were used. Furthermore, for MOVIE, only the LIVE data set and the IT-IST data set, excluding the sequences *Football* and *Tempete*, were used, as the available MOVIE implementation was unable to process all sequences in the TUM1080p25 and TUM1080p50 data set, and the two excluded sequences in the IT-IST data set.

Except for MOVIE that already provides a visual quality prediction for the complete video sequence, the prediction results of the individual frames were temporally pooled into an overall visual quality prediction by averaging over all frames. For SSIM, IW-SSIM and MOVIE the implementations' default settings for the metrics' parameters were used, that are based on the suggested values in the original contributions. For NIQE it is recommended to set the patch size of the NIQE algorithm to the same size as the subdivisions in the images and thus the patch size was set to 16, corresponding to the macroblock size for H.264/AVC encoded video. Furthermore, the default model calibrated on a set of 125 undistorted images provided with the NIQE reference implementation was used. Additionally, all metrics were only applied to the 8 bit luma component of the frames.

Even tough I rejected data fitting in the performance comparison of the data analysis based example metrics in the previous section, there are two valid reasons to apply it to the comparison metrics: firstly, unlike for the data analysis based metrics, the visual quality predictions of the comparison metrics are not necessarily in the same interval $[0, 1]$ as the visual quality $\mathbf{y}$ and thus results for RMSE, RMSE_E and OR are not interpretable. Secondly data fitting was an integral part of the original contributions describing SSIM, IW-SSIM, NIQE and MOVIE, and thus it may disadvantage the metrics in the comparison if the fitting is omitted.

In the original contribution, SSIM and MOVIE were fitted to the visual quality using a four parameter logistic function suggested by VQEG in the FRTV Phase I project [328], and IW-SSIM and NIQE were fitted to the visual quality using a the six parameter logistic function suggested by Sheikh et al. [289]. In the second phase of the FRTV project [329], however, VQEG noted that the fitting process for logistic functions with more than three parameters often didn't converge. Consequently, VQEG replaced the logistic function in the following HDTV project [330] by a monotonic cubic polynomial function. Additionally, the use of a third degree polynomial function has also been recommended recently in ITU-T P.1401 [110]

Considering these issues, I decided to fit only a linear (polynomial) function, aimed mainly at aligning the ranges of the different scales for the prediction quality, affecting the RMSE, RMSE_E and OR, but not the correlation metrics. Although this may disadvantage some of the comparison metrics, this is a more realistic application scenario for video quality metrics as already discussed in Section 9.1.2. Still, as an indicator of the best possible prediction performance achieved by the comparison metrics with respect to the used data sets, albeit under unrealistic conditions, I also fitted a cubic polynomial function as suggested by VQEG and in ITU-T P.1401, but the corresponding results are not used in the direct comparison and only provided for information in the Appendix. The fitting was done with the `fit` function of the MATLAB R2012b Curvefitting Toolbox [204], and it is interesting to note that similar problems as in [329] with respect to the convergence of the four and five parameter logistic functions were also encountered.

Presentation of the results Similar to the comparison of the data analyis methods in the previous section, only scatter plots for the IT-IST data set are provided in this section, as it is the largest and content-wise most comprehensive data set. For the TUM1080p25, TUM1080p50 and LIVE data set, scatter plots are provided in Appendix B.7. Also the OR, RMSE and RMSE_E are expressed again as the complement to their maximum value.

Prediction performance The comparision of the prediction performance of the bitstream-based example metric built with trilinear PLSR with the selected state-of-the-art metrics is shown for all data sets in Fig 9.15 on the next page and additionally for the IT-IST data set as a scatter plot in Fig. 9.16 on page 176. All state-of-the-art metrics were fit to the data with a linear function. The results of this evaluation are also available in Table B.8 in the Appendix.

The bitstream-based example metric built with trilinear PLSR outperforms all the state-of-the-art metrics used in the comparison for the IT-IST and TUM1080p25 data set, and for the TUM1080p50 data set the example metric is only outperformed by the full-reference IW-SSIM as shown in Fig. 9.15. In particular, the only no-reference metric in the comparison, NIQE, is outperformed for the IT-IST, TUM1080p25 and TUM1080p50 data sets and only for the LIVE data set NIQE has a slight advantage with respect to the RMSE.

For the LIVE data set, we can observe that compared to the example metric and NIQE,

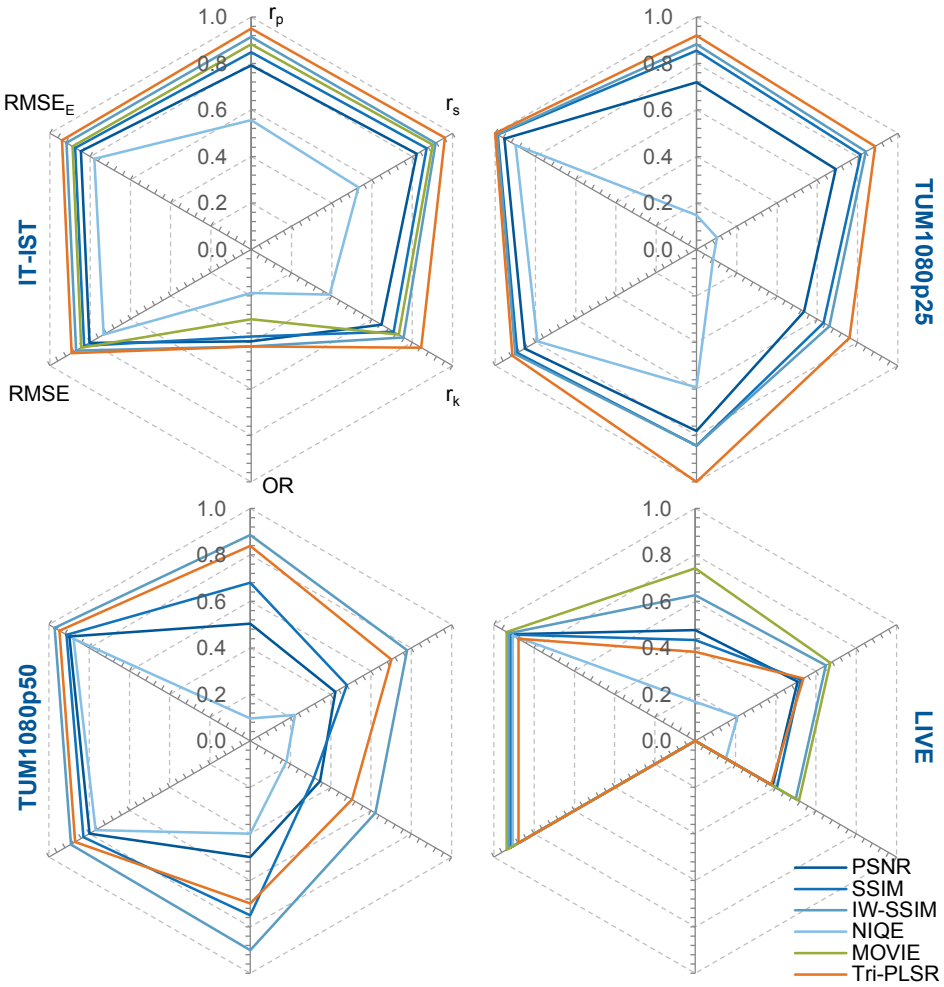

Figure 9.15.: Prediction performance for the bitstream-based example metric built with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE metrics with respect to the Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive RMSE_E. For clarity only the axes for the IT-IST data set are labelled

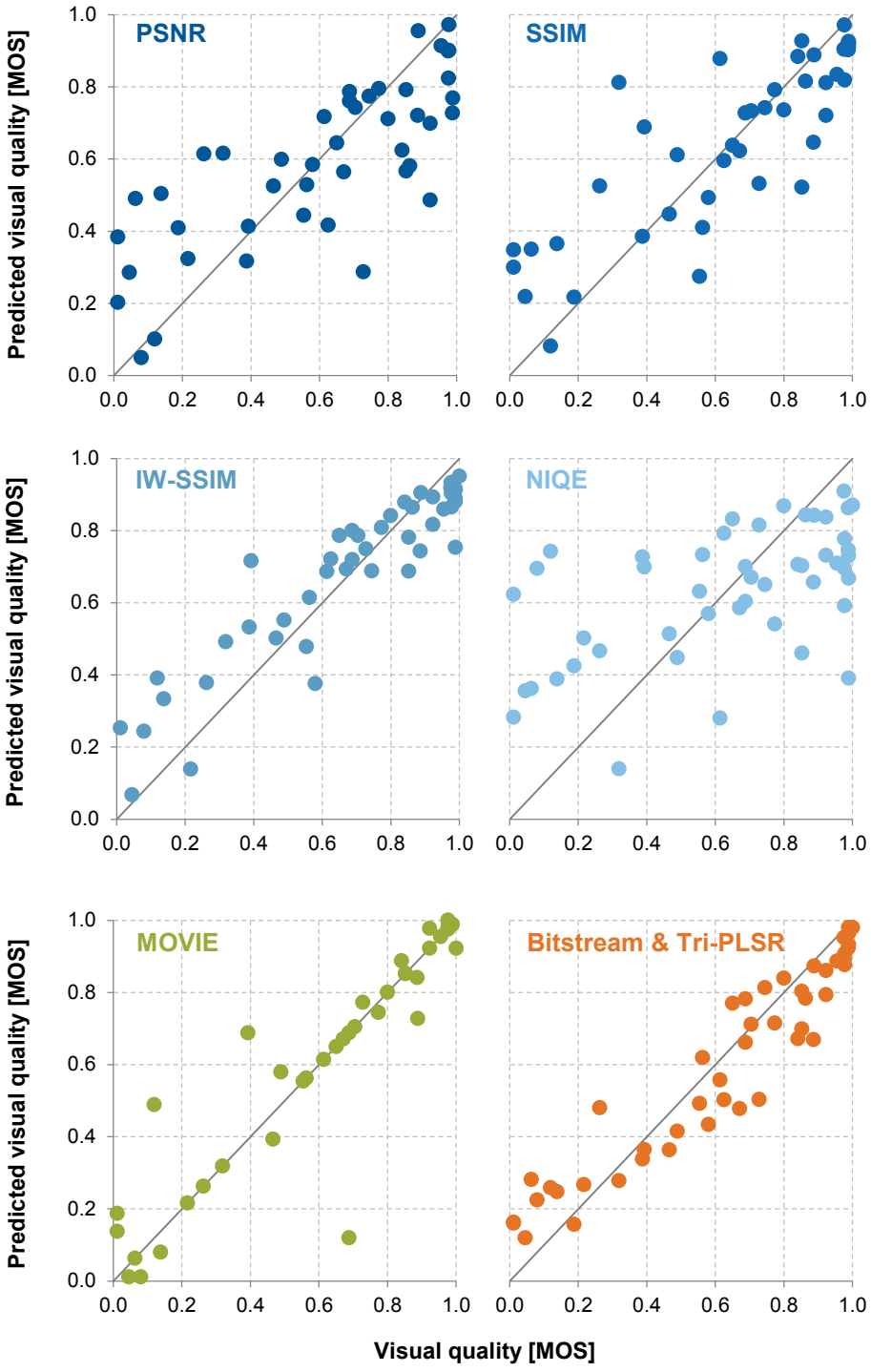

Figure 9.16.: Scatter plots of the prediction results of the bitstream-based example metric built with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE for the IT-IST data set, showing the visual quality y against the predicted visual quality $\widehat{y}$

all other metrics have significantly better prediction performance. Considering that NIQE utilises the statistical properties of the frames and in particular the standard deviation to fit a multivariate Gaussian model, it stands to reason that NIQE also suffers from the lack of variation within the LIVE data set, supporting the argument in the previous section that the lack of achieved prediction performance with the data analysis based metric for the LIVE data set may be caused by a lack for variation within the features extracted from the sequences.

Comparing the state-of-the-art metrics against each other, IW-SSIM is for most data sets the best performing metric, even outperforming the dedicated video quality metric MOVIE except for the LIVE data set. Also not surprisingly IW-SSIM as the current pinnacle of the SSIM family of metrics outperforms both SSIM and PSNR, as can be observed in the scatter plot for the IT-IST data set in Fig. 9.16. Slightly disappointing is the prediction performance of NIQE, especially considering it achieved a Pearson correlation of > 0.9 in [208]. But as NIQE is calibrated on a set of undistorted images, it may very well be that the still images used to train the general model are not representative enough of typical content in video sequences' frames.

The difference in the prediction results of the example metric and PSNR, SSIM, NIQE and MOVIE with respect to the Pearson correlation, Spearman rank correlation and RMSE is statistical significant for the IT-IST data set at the 0.05 level with a p-value of $p < 0.05$ and only for the IW-SSIM no statistical significant difference could be confirmed for this data set. For the other data sets, no general trend could be observed and the complete results of the significance testing are provided in Fig. B.20 in the Appendix.

Applying cubic data fitting instead of linear data fitting does not change the overall ranking of the state-of-the-art metrics in relation to the example metric and the only difference is that the prediction performance increased e.g. for IW-SSIM from a Pearson correlation of 0.91 to 0.94, and I refer to Appendix B.8 for the results with cubic data fitting. In particular, the statistical significance of the superiority of the bitstream-based example metric built with trilinear PLSR for the IT-IST data set is maintained even with the cubic data fitting as illustrated in Fig. B.26 in the Appendix. Therefore it can be assumed that only using linear fitting did not penalise the state-of-the-art metrics in the comparison.

9.5. Summary

In this chapter, two different example video quality metrics both representing the two main concepts in the engineering design approach to video quality metrics were used to evaluate if multi-way data analysis methods really provide an advantage compared to two-way data analysis methods combined with temporal pooling in the context of the design of video quality prediction models. This assessment was done using four different data sets, covering a wide range of content and resolutions encountered in real-life applications.

Two-way versus multi-way data analysis Excluding the results for the LIVE data set, for both example metrics the results show that the multi-way approach to data analysis outperforms pooling and two-way data analysis, supporting my arguments in the previous chapters that considering video in its natural three-way structure without pooling allows us to build better prediction models. In particular, the trilinear PLSR outperforms all other data analysis methods in five out of six metric and data set combinations, indicating that a three-way component model is the best way to describe the latent variables contributing to the visual quality prediction. This is also supported by the fact that although fitting a two-way component model to three-way data with unfolding and two-way data analysis methods results in a better prediction performance than two-way data analysis methods and pooling of the three-way data, it is still inferior to the trilinear PLSR.

One outlier to this trend is the LIVE data set, where for both example metrics two-way methods are equal or better than the multi-way methods and in particular the extremely good performance of the MLR for the pixel-based example metrics is surprising. This, however, is very likely to be caused by the observed apparent lack of variance within the LIVE data set, as indicated by the general under performance of the component-based methods for this data set.

Multi-way data analysis design approach versus state-of-the-art Comparing the best performing example metric designed with multi-way data analysis to the state-of-the-art, I have shown that excluding the problematic LIVE data set, the bitstream-based example metric build with trilinear PLSR outperforms metrics in literature for two out of three data sets. In particular, it is even superior to full-reference metrics that utilise significant more information in their prediction models. Also it is important to note that unlike in the design process of other video quality metrics, no assumptions about underlying dependencies between the extracted features are made, but that the design is purely data-driven and not optimised in any way for certain desired properties.

Considering then the example metrics as proper video quality metrics and using trilinear PLSR as the best performing method for building both example metrics, the compliance of the example metrics with respect to the requirements on video quality metrics defined in Chapter 3 can be determined and is shown in Table 9.2 on the next page. For the bitstream-based example metric, all requirements are fulfilled, whereas the non-optimised

Table 9.2.: Compliance with the requirements in each category for example metrics

Name	Compliance				Type
	Temporal	Reference	Prediction	Validation	
Bitstream-based example metric					
Trilinear PLSR	●●	●●	●●	●●	VQ
Pixel-based example metric					
Trilinear PLSR	●●	●●	○●	●●	VQ
Trilinear PLSR, optimised	●●	●●	●●	●●	VQ

compliance level: ●●= full, ○●= acceptable, ○○= insufficient; – = no information available

version of the pixel-based example metrics only fulfils three of the requirements fully. This reflects the overall trend in video quality metrics that the more specialised bitstream-based metrics have a better prediction performance than the more general pixel-based metrics. But using the optimised version of the pixel-based metric that takes non-linearities and the importance of features into account, this optimised pixel-based example metric also fulfils all requirements fully.

10. Conclusion

In this thesis, I introduced the data driven design methodology with multi-way data analysis to the design of video quality metrics. Using two simple example metrics designed with the proposed approach, I demonstrated the advantages of utilising the natural three-way structure of video by applying multi-way data analysis methods that exploit this three-way structure in the model building process. Moreover, the metrics designed with the proposed approach are not only competitive with existing video quality metrics in the state-of-the-art, but also outperform the metrics in the state-of-the-art. This is particularly impressive as the prediction models were gained purely by analysing the data during the training phase, unlike many existing metrics that often use sophisticated models describing partial aspects of human perception.

Moving beyond the scope of this thesis, the proposed method can be utilised for building video quality metrics using arbitrary features, but may even be adapted to other, more efficient model building methods, including, but not limited to possible multi-way machine learning methods in the future. Similarly, additional modes may be included into the prediction process, representing additional views, different spectral domains, or any other quantifiable domain of video sequences that can be considered as an additional, but distinct mode.

The idea of applying multi-way data analysis methods to data exhibiting a multi-way structure itself is nothing new and has already been used frequently in chemometrics applications. Yet, this powerful concept of multi-way algorithms and methods is largely unknown in the signal processing research community in general, and the image and video processing research community in particular, where most methods and algorithms are still focused on matrices or two-way arrays as the essential structure of the data.

This thesis demonstrates that by simply looking beyond one's own research area into at first glance completely unrelated fields, a wealth of new methods and concepts can be gained to solve existing problems in one's own research area, especially as although often covering very different topics, still similar underlying basic problems are encountered in many fields. However, too often such opportunities are missed due to a narrow focus on the own research area and I conclude this thesis therefore with a remark by the late Sijmen de Jong [132]

> It is most unfortunate that there is such a large scatter of data analytic communities (applied statistics, machine learning, and all the –ometrics subdisciplines) with each faction so reluctant to learn from the other ones.
>
> The waste of effort!

Bibliography

1. R. Aldridge, J. Davidoff, M. Ghanbari, D. Hands, and D. Pearson. Recency effect in the subjective assessment of digitally-coded television pictures. In *Image Processing and its Applications, 1995. Fifth International Conference on*, pp. 336–339. July 1995. ISBN 0-85296-642-3. doi:10.1049/cp:19950676.

2. D.M. Allen. The relationship between variable selection and data agumentation and a method for prediction. In *Technometrics*, 16(1), pp. 125–127, February 1974. ISSN 1537-2723. doi:10.1080/00401706.1974.10489157.

3. E. Anderssen, K. Dyrstad, F. Westad, and H. Martens. Reducing over-optimism in variable selection by cross-model validation. In *Chemometrics and Intelligent Laboratory Systems*, 84(1-2), pp. 69–74, 2006. ISSN 0169-7439. doi:10.1016/j.chemolab.2006.04.021.

4. C.A. Andersson and R. Bro. Improving the speed of multi-way algorithms: Part I. Tucker3. In *Chemometrics and Intelligent Laboratory Systems*, 42(1–2), pp. 93–103, August 1998. ISSN 0169-7439. doi:10.1016/S0169-7439(98)00010-0.

5. C.A. Andersson and R. Bro. The n-way toolbox for MATLAB. In *Chemometrics and Intelligent Laboratory Systems*, 52(1), pp. 1–4, August 2000. ISSN 0169-7439. doi:10.1016/S0169-7439(00)00071-X.

6. F.J. Anscombe. Graphs in statistical analysis. In *The American Statistician*, 27(1), pp. 17–21, February 1973. ISSN 0003-1305.

7. S. Argyropoulos, A. Raake, M.N. Garcia, and P. List. No-reference bit stream model for video quality assessment of H.264/AVC video based on packet loss visibility. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 1169–1172. May 2011. ISBN 978-1-4577-0537-3. ISSN 1520-6149. doi:10.1109/ICASSP.2011.5946617.

8. S. Argyropoulos, A. Raake, M.N. Garcia, and P. List. No-reference video quality assessment for SD and HD H.264/AVC sequences based on continuous estimates of packet loss visibility. In *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on*, pp. 31–36. September 2011. ISBN 978-1-4577-1334-7. doi:10.1109/QoMEX.2011.6065708.

9. D. Ariely. Combining experiences over time: the effects of duration, intensity changes and on-line measurements on retrospective pain evaluations. In *Journal of Behavioral Decision Making*, 11(1), pp. 19–45, March 1998. ISSN 1099-0771. doi: 10.1002/(SICI)1099-0771(199803)11:1<19::AID-BDM277>3.0.CO;2-B.

10. S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. In *Statistics Surveys*, 4, pp. 40–79, 2010. ISSN 1935-7516. doi:10.1214/09-SS054.

11. M. Barkowsky, J. Bialkowski, B. Eskofier, R. Bitto, and A. Kaup. Temporal trajectory aware video quality measure. In *Selected Topics in Signal Processing, IEEE Journal of*, 3(2), pp. 266–279, April 2009. ISSN 1932-4553. doi:10.1109/JSTSP.2009.2015375.

12. V. Baroncini. New tendencies in subjective video quality evaluation. In *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 89(11), pp. 2933–2937, November 2006. ISSN 1745-1337.

13. D. Belsley, E. Kuh, and R. Welsch. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. John Wiley & Sons, Inc. New York, 2004. ISBN 0-471-69117-8. doi:10.1002/0471725153.

14. J. ten Berge. *Least Squares Optimization in Multivariate Analysis*. DSWO Press, Leiden University, 1993.

15. J. ten Berge and H. Kiers. Convergence properties of an iterative procedure of ipsatizing and standardizing a data matrix, with applications to parafac/candecomp preprocessing. In *Psychometrika*, 54(2), pp. 231–235, June 1989. ISSN 0033-3123. doi:10.1007/BF02294517.

16. T. Borer and T. Davies. *Dirac - video compression using open technology*. Technical Report WHP 117, BBC Research & Development, July 2005.

17. T. Borer, T. Davies, and A. Suraparaju. *Dirac Video Compression*. Technical Report WHP 124, BBC Research & Development, September 2005.

18. F. Boulos, W. Chen, B. Parrein, and P. Le Callet. Region-of-interest intra prediction for H.264/AVC error resilience. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pp. 3109–3112. November 2009. ISBN 978-1-4244-5655-0. ISSN 1522-4880. doi:10.1109/ICIP.2009.5414458.

19. G. Box, W. Hunter, and J. Hunter. *Statistics for Experimenters. An introduction to design, data analysis and model building.* John Wiley & Sons, Ltd., New York, 1978. ISBN 978-0-471-71813-0.

20. R.A. Bradley. Science, statistics, and paired comparisons. In *Biometrics*, 32(2), pp. 213–239, June 1976. ISSN 0006-341X.

21. C.J. van den Branden Lambrecht and O. Verscheure. Perceptual quality measure using a spatiotemporal model of the human visual system. In V. Bhaskaran, F. Sijstermans, and S. Panchanathan (eds.), *Digital Video Compression: Algorithms and Technologies*, pp. 450–461. January 1996. doi:10.1117/12.235440.

22. T. Brandão and M.P. Queluz. No-reference image quality assessment based on DCT domain statistics. In *Signal Processing*, 88(4), pp. 822–833, March 2008. ISSN 0165-1684. doi:10.1016/j.sigpro.2007.09.017.

23. T. Brandão and M.P. Queluz. No-reference quality assessment of H.264/AVC encoded video. In *Proceedings of International Workshop on Video Processing and Quality Metrics for Consumer Electronics - VPQM*, pp. 51–56. January 2010.

24. T. Brandão and M.P. Queluz. No-reference quality assessment of H.264/AVC encoded video. In *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(11), pp. 1437–1447, November 2010. ISSN 1051-8215. doi:10.1109/TCSVT.2010.2077474.

25. T. Brandão, L.R. Roque, and M.P. Queluz. Quality assessment of H.264/AVC encoded video. In *Proceedings of Conference on Telecommunications - ConfTele*. April 2009.

26. L. Breiman and P. Spector. Submodel selection and evaluation in regression. the x-random case. In *International Statistical Review / Revue Internationale de Statistique*, 60(3), pp. 291–319, December 1992. ISSN 0306-7734.

27. M. Brill and J. Lubin. *Final Report: Sarnoff JND Vision Model for Flat-Panel Design*. Technical Report NAS2-14257, Sarnoff Cooperation, May 1998.

28. R. Bro. *Multi-way Analysis in the Food Industry - Models, Algorithms, and Applications*. Ph.D. thesis, University of Amsterdam, Amsterdam, November 1998.

29. R. Bro. Multiway calibration. Multilinear PLS. In *Journal of Chemometrics*, 10(1), pp. 47–61, January 1996. ISSN 1099-128X. doi:10.1002/(SICI)1099-128X(199601)10:1<47::AID-CEM400>3.0.CO;2-C.

30. R. Bro. PARAFAC. Tutorial and applications. In *Chemometrics and Intelligent Laboratory Systems*, 38(2), pp. 149–171, October 1997. ISSN 0169-7439. doi:10.1016/S0169-7439(97)00032-4.

31. R. Bro and C.A. Andersson. Improving the speed of multiway algorithms: Part ii: Compression. In *Chemometrics and Intelligent Laboratory Systems*, 42(1–2), pp. 105–113, August 1998. ISSN 0169-7439. doi:10.1016/S0169-7439(98)00011-2.

32. R. Bro and A.K. Smilde. Centering and scaling in component analysis. In *Journal of Chemometrics*, 17(1), pp. 16–33, January 2003. ISSN 1099-128X. doi:10.1002/cem.773.

33. R. Bro, A.K. Smilde, and S. de Jong. On the difference between low-rank and subspace approximation: improved model for multi-linear PLS regression. In *Chemometrics and Intelligent Laboratory Systems*, 58(1), pp. 3–13, September 2001. ISSN 0169-7439. doi:10.1016/S0169-7439(01)00134-4.

34. D. Brunet, E. Vrscay, and Z. Wang. On the mathematical properties of the structural similarity index. In *Image Processing, IEEE Transactions on*, 21(4), pp. 1488–1499, April 2012. ISSN 1057-7149. doi:10.1109/TIP.2011.2173206.

35. A.J. Burnham, J.F. MacGregor, and R. Viveros. Latent variable multivariate regression modeling. In *Chemometrics and Intelligent Laboratory Systems*, 48(2), pp. 167–180, August 1999. ISSN 0169-7439. doi:10.1016/S0169-7439(99)00018-0.

36. S.L. Campbell and C.D. Meyer. *Generalized Inverses of Linear Transformations*. Society for Industrial and Applied Mathematics, Philadelphia, 2009. ISBN 978-0-89871-904-8. doi:10.1137/1.9780898719048.

37. J. Carroll and J.J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. In *Psychometrika*, 35(3), pp. 283–319, September 1970. ISSN 0033-3123. doi:10.1007/BF02310791.

38. R. Cattell. "Parallel proportional profiles" and other principles for determining the choice of factors by rotation. In *Psychometrika*, 9(4), pp. 267–283, December 1944. ISSN 0033-3123. doi:10.1007/BF02288739.

39. R. Cattell. The three basic factor-analytic research designs–their interrelations and derivatives. In *Psychological Bulletin*, 49(5), pp. 499–520, September 1952. ISSN 0033-3123. doi:10.1007/BF02288739.

40. D. Chandler and S. Hemami. VSNR: A wavelet-based visual signal-to-noise ratio for natural images. In *Image Processing, IEEE Transactions on*, 16(9), pp. 2284–2298, September 2007. ISSN 1057-7149. doi:10.1109/TIP.2007.901820.

41. C. Charrier, K. Knoblauch, L. Maloney, A. Bovik, and A. Moorthy. Optimizing multiscale SSIM for compression via MLDS. In *Image Processing, IEEE Transactions on*, 21(12), pp. 4682–4694, December 2012. ISSN 1057-7149. doi:10.1109/TIP.2012.2210723.

42. M.J. Chen and A. Bovik. No-reference image blur assessment using multiscale gradient. In *EURASIP Journal on Image and Video Processing*, 1(1), pp. 1–11, July 2011. ISSN 1687-5281. doi:10.1186/1687-5281-2011-3.

43. C. Cheng and H. Wang. Quality assessment for color images with tucker decomposition. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pp. 1489–1492. September 2012. ISBN 978-1-4673-2532-5. ISSN 1522-4880. doi: 10.1109/ICIP.2012.6467153.

44. S. Chikkerur, V. Sundaram, M. Reisslein, and L. Karam. Objective video quality assessment methods: A classification, review, and performance comparison. In *Broadcasting, IEEE Transactions on*, 57(2), pp. 165–182, June 2011. ISSN 0018-9316. doi:10.1109/TBC.2011.2104671.

45. 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, Chichester, 2009. ISBN 978-0-470-74666-0.

46. R.D. Cook and S. Weisberg. *Residuals and Influence in Regression*. Chapman and Hall, New York, 1982. ISBN 0-412-24280-0.

47. P. Corriveau, C. Gojmerac, B. Hughes, and L. Stelmach. All subjective scales are not created equal: The effects of context on different scales. In *Signal Processing*, 77(1), pp. 1–9, August 1999. ISSN 0165-1684. doi:10.1016/S0165-1684(99)00018-3.

48. S.J. Daly. *Digital Images and Human Vision*, chapter The visible differences predictor: an algorithm for the assessment of image fidelity., pp. 179–206. MIT Press, Cambridge, MA, 1993. ISBN 0-262-23171-9.

49. B.S. Dayal and J.F. MacGregor. Improved PLS algorithms. In *Journal of Chemometrics*, 11(1), pp. 73–85, January 1997. ISSN 1099-128X. doi:10.1002/(SICI)1099-128X(199701)11:1<73::AID-CEM435>3.0.CO;2-\#.

50. A. Eden. No-reference estimation of the coding PSNR for H.264-coded sequences. In *Consumer Electronics, IEEE Transactions on*, 53(2), pp. 667–674, May 2007. ISSN 0098-3063. doi:10.1109/TCE.2007.381744.

51. B. Efron. *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics, Philadelphia, 1982. ISBN 978-1-61197-031-9. doi:10.1137/1.9781611970319.

52. K. Egiazarian, J. Astola, N. Ponomarenko, V. Lukin, F. Battisti, and M. Carli. Two new full-reference quality metrics based on HVS. In *Second International Workshop on Video Processing and Quality Metrics for Consumer Electronics - VPQM*. January 2006.

53. L. Eldén and B. Savas. A newton-grassmann method for computing the best multilinear rank-(r_1, r_2, r_3) approximation of a tensor. In *SIAM Journal on Matrix Analysis and Applications*, 31(2), pp. 248–271, March 2009. ISSN 1095-7162. doi: 10.1137/070688316.

54. U. Engelke, M. Barkowsky, P. Le Callet, and H.J. Zepernick. Modelling saliency awareness for objective video quality assessment. In *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, pp. 212–217. June 2010. ISBN 978-1-4244-6959-8. doi:10.1109/QOMEX.2010.5516159.

55. ETSI. *Human Factors (HF); Quality of Experience (QoE) requirements for real-time communication services*. Technical Report TR 102 643, V1.0.1, ETSI, December, 2009.

56. N.K.M. Faber, R. Bro, and P.K. Hopke. Recent developments in CANDECOMP/PARAFAC algorithms: a critical review. In *Chemometrics and Intelligent Laboratory Systems*, 65(1), pp. 119–137, January 2003. ISSN 0169-7439. doi: 10.1016/S0169-7439(02)00089-8.

57. M.C.Q. Farias, M. Carli, and S. Mitra. Objective video quality metric based on data hiding. In *Consumer Electronics, IEEE Transactions on*, 51(3), pp. 983–992, August 2005. ISSN 0098-3063. doi:10.1109/TCE.2005.1510512.

58. M.C.Q. Farias and S. Mitra. No-reference video quality metric based on artifact measurements. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pp. III–141–4. September 2005. ISBN 0-7803-9134-9. doi: 10.1109/ICIP.2005.1530348.

59. O.D. Faugeras. Digital color image processing within the framework of a human visual model. In *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(4), pp. 380–393, August 1979. ISSN 0096-3518. doi:10.1109/TASSP.1979.1163262.

60. X. Feng, T. Liu, D. Yang, and Y. Wang. Saliency based objective quality assessment of decoded video affected by packet losses. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 2560–2563. October 2008. ISBN 978-1-4244-1764-3. doi:10.1109/ICIP.2008.4712316.

61. G. Ferguson. The concept of parsimony in factor analysis. In *Psychometrika*, 19(4), pp. 281–290, December 1954. ISSN 0033-3123. doi:10.1007/BF02289228.

62. E.C. Fieller, H.O. Hartley, and E.S. Pearson. Tests for rank correlation coefficients. I. In *Biometrika*, 44(3/4), pp. 470–481, December 1957. ISSN 0006-3444.

63. R.A. Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. In *Biometrika*, 10(4), pp. 507–521, May 1915. ISSN 0006-3444.

64. R.A. Fisher and W.A. Mackenzie. Studies in crop variation. II. The manurial response of different potato varieties. In *The Journal of Agricultural Science*, 13(3), pp. 311–320, June 1923. ISSN 1469-5146. doi:10.1017/S0021859600003592.

65. K. Fliegel. *List of Qualinet Multimedia Databases v3.0*. Technical Report Qo0207, Qualinet - European Network on Quality of Experience in Multimedia Systems and Services, February 2012.

66. K. Fliegel. *Qualinet Multimedia Databases v3.0*. Technical Report Qo0206, Qualinet - European Network on Quality of Experience in Multimedia Systems and Services, February 2012.

67. I.E. Frank and J.H. Friedman. A statistical view of some chemometrics regression tools. In *Technometrics*, 35(2), pp. 109–135, May 1993. ISSN 0040-1706.

68. B.L. Fredrickson. Extracting meaning from past affective experiences: The importance of peaks, ends, and specific emotions. In *Cognition & Emotion*, 14(4), pp. 577–606, 2000. ISSN 0269-9931. doi:10.1080/026999300402808.

69. B.L. Fredrickson and D. Kahneman. Duration neglect in retrospective evaluations of affective episodes. In *Journal of Personality and Social Psychology*, 65(1), pp. 45–55, July 1993. ISSN 0022-3514. doi:10.1037/0022-3514.65.1.45.

70. P. Gastaldo, G. Parodi, J. Redi, and R. Zunino. No-reference quality assessment of JPEG images by using CBP neural networks. In J. Sá, L. Alexandre, W. Duch, and D. Mandic (eds.), *Artificial Neural Networks – ICANN 2007*, volume 4669 of *Lecture Notes in Computer Science*, pp. 564–572. Springer, Berlin & Heidelberg, 2007. ISBN 978-3-540-74693-5. doi:10.1007/978-3-540-74695-9_58.

71. D. Geerts, K. De Moor, I. Ketykó, A. Jacobs, J. Van den Bergh, W. Joseph, L. Martens, and L. De Marez. Linking an integrated framework with appropriate methods for measuring QoE. In *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, pp. 158–163. June 2010. ISBN 978-1-4244-6959-8. doi:10.1109/QOMEX.2010.5516292.

72. S. Geisser. The predictive sample reuse method with applications. In *Journal of the American Statistical Association*, 70(350), pp. 320–328, June 1975. ISSN 1537-274X. doi:10.1080/01621459.1975.10479865.

73. B. Girod. *Digital Images and Human Vision*, chapter What's wrong with mean-squared error?, pp. 207–220. MIT Press, Cambridge, 1993. ISBN 0-262-23171-9.

74. G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. In *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis*, 2(2), pp. 205–224, 1965. ISSN 0887-459X. doi:10.1137/0702016.

75. G. Golub and C. Reinsch. Singular value decomposition and least squares solutions. In *Numerische Mathematik*, 14(5), pp. 403–420, April 1970. ISSN 0029-599X. doi:10.1007/BF02163027.

76. G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, 3rd edition. Johns Hopkins University Press, Baltimore, 1996. ISBN 0-8018-5413-X.

77. M. Goudarzi, L. Sun, and E. Ifeachor. Audiovisual quality estimation for video calls in wireless applications. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5. December 2010. ISSN 1930-529X. doi:10.1109/GLOCOM.2010.5683109.

78. J. Guo, M. Van Dyke-Lewis, and H. Myler. Gabor difference analysis of digital video quality. In *Broadcasting, IEEE Transactions on*, 50(3), pp. 302–311, September 2004. ISSN 0018-9316. doi:10.1109/TBC.2004.834020.

79. L. Haglund. *The SVT Multi Format Test Set Version 1.0*. Technical Report, Sveriges Television (SVT), February 2006.

80. D.S. Hands and S.E. Avons. Recency and duration neglect in subjective assessment of television picture quality. In *Applied Cognitive Psychology*, 15(6), pp. 639–657, November 2001. ISSN 1099-0720. doi:10.1002/acp.731.

81. P. Hanhart. VQMT: video quality measurement tool. Online, March 2013. URL `http://mmspg.epfl.ch/vqmt`, accessed on 28.06.2013. Version 1.1.

82. R.A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. In *UCLA Working Papers in Phonetics*, 16(16), pp. 1–84, December 1970.

83. R.A. Harshman and M.E. Lundy. Data preprocessing and the extended PARAFAC model. In H.G. Law, J. C. W. Snyder, J. Hattie, and R.P. McDonald (eds.), *Research methods for multimode data analysis*, pp. 216–284. Praeger, New York, 1984. ISBN 0-03-062826-1.

84. I.S. Helland. Partial least squares regression and statistical models. In *Scandinavian Journal of Statistics*, 17(2), pp. 97–114, 1990. ISSN 0303-6898.

85. I.S. Helland and T. Almoy. Comparison of prediction methods when only a few components are relevant. In *Journal of the American Statistical Association*, 89(426), pp. 583–591, June 1994. ISSN 0162-1459.

86. F.L. Hitchcock. The expression of a tensor or a polyadic as a sum or products. In *Journal of Mathematics and Physics*, 6, pp. 164–189, 1927.

87. F.L. Hitchcock. Multiple invariants and generalized rank of a p-way matrix or tensor. In *Journal of Mathematics and Physics*, 7, pp. 39–79, 1927.

88. D.C. Hoaglin and R.E. Welsch. The hat matrix in regression and ANOVA. In *The American Statistician*, 32(1), pp. 17–22, February 1978. ISSN 1537-2731. doi:10.1080/00031305.1978.10479237.

89. R.M. Hogarth and H.J. Einhorn. Order effects in belief updating: The belief-adjustment model. In *Cognitive Psychology*, 24(1), pp. 1–55, January 1992. ISSN 0010-0285. doi:http://dx.doi.org/10.1016/0010-0285(92)90002-J.

90. H. Hotelling. Analysis of a complex of statistical variables into principal components. In *Journal of educational psychology*, 24(6), pp. 417–441, September 1933. ISSN 0022-0663. doi:10.1037/h0071325.

91. A. Höskuldsson. PLS regression methods. In *Journal of Chemometrics*, 2(3), pp. 211–228, June 1988. ISSN 1099-128X. doi:10.1002/cem.1180020306.

92. A. Höskuldsson. Quadratic PLS regression. In *Journal of Chemometrics*, 6(6), pp. 307–334, November 1992. ISSN 1099-128X. doi:10.1002/cem.1180060603.

93. J. Håstad. Tensor rank is NP-complete. In *Journal of Algorithms*, 11(4), pp. 644–654, December 1990. ISSN 0196-6774. doi:10.1016/0196-6774(90)90014-6.

94. Q. Huynh-Thu, M.N. Garcia, F. Speranza, P. Corriveau, and A. Raake. Study of rating scales for subjective quality assessment of high-definition video. In *Broadcasting, IEEE Transactions on*, 57(1), pp. 1–14, March 2011. ISSN 0018-9316. doi:10.1109/TBC.2010.2086750.

95. Instituto de Telecomunicações at Instituto Superior Técnico, Lisbon. Quality assessment of H.264/AVC encoded sequences. Online. URL `http://amalia.img.lx.it.pt/~tgsb/H264_test/`, accessed on 21.06.2013.

96. *Information technology – Coding of audio-visual objects – Part 2: Visual*. Standard ISO/IEC 14496-2, revision 3. ISO/IEC, June 2004.

97. *Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*. Standard ISO/IEC 14496-10, revision 7. ISO/IEC, March 2012.

98. *Information technology – Generic coding of moving pictures and associated audio information: Video*. Standard ISO/IEC 13818-2, revision 2. ISO/IEC, March 2012.

99. ITU-R. *Studies toward the unification of picture assessment methodology*. Technical Report BT.1082-1, ITU-R, January 1990.

100. *Subjective assessment methods for image quality in high-definition television*. Standard ITU-R BT.710, revision 4. ITU-R, November 1998.

101. *Subjective assessment of standard definition digital television (SDTV) systems*. Standard ITU-R BT.1129, revision 2. ITU-R, February 1998.

102. *Parameter values for the HDTV standards for production and international programme exchange*. Standard ITU-R BT.709, revision 5. ITU-R, April 2002.

103. *Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference*. Standard ITU-R BT.1683, revision 1. ITU-R, June 2004.

104. *Methodology for the subjective assessment of video quality in multimedia applications*. Standard ITU-R BT.1788, revision 1. ITU-R, January 2007.

105. *Objective perceptual video quality measurement techniques for broadcasting applications using HDTV in the presence of a full reference signal*. Standard ITU-R BT.1866, revision 1. ITU-R, March 2010.

106. *Objective perceptual visual quality measurement techniques for broadcasting applications using low definition television in the presence of a reduced bandwidth reference*. Standard ITU-R BT.1867, revision 1. ITU-R, March 2010.

107. *Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios*. Standard ITU-R BT.601, revision 3. ITU-R, March 2011.

108. *General viewing conditions for subjective assessment of quality of SDTV and HDTV television pictures on flat panel displays*. Standard ITU-R BT.2022, revision 1. ITU-R, August 2012.

109. *Methodology for the Subjective Assessment of the Quality for Television Pictures*. Standard ITU-R BT.500, revision 13. ITU-R, January 2012.

110. *Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models*. Standard ITU-T P.1401, revision 1. ITU-T, July 2012.

111. *Objective perceptual video quality measurement techniques for broadcasting applications using HDTV in the presence of a full reference signal*. Standard ITU-R BT.1907, revision 1. ITU-R, January 2012.

112. *Objective video quality measurement techniques for broadcasting applications using HDTV in the presence of a reduced reference signal*. Standard ITU-R BT.1908, revision 1. ITU-R, January 2012.

113. *Test materials to be used in assessment of picture quality*. Standard ITU-R BT.1210, revision 4. ITU-R, January 2012.

114. *A reference viewing environment for evaluation of HDTV program material or completed programmesI*. Standard ITU-R BT.[REF-VIEW], revision 1. ITU-R, May 2013. Draft.

115. *Information technology – Generic coding of moving pictures and associated audio information: Video*. Standard ITU-T H.262, revision 2. ITU-T, February 2000.

116. *Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference*. Standard ITU-T J.144, revision 2. ITU-T, March 2004.

117. *Video coding for low bit rate communication*. Standard ITU-T H.263, revision 2. ITU-T, January 2005.

118. *Definitions of terms related to quality of service*. Standard ITU-T E.800, revision 5. ITU-T, September 2008.

119. *Objective perceptual multimedia video quality measurement in the presence of a full reference*. Standard ITU-T J.247, revision 1. ITU-T, August 2008.

120. *Perceptual visual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference*. Standard ITU-T J.246, revision 1. ITU-T, August 2008.

121. *Subjective video quality assessment methods for multimedia applications*. Standard ITU-T P.910, revision 3. ITU-T, April 2008.

122. *Vocabulary for performance and quality of service - Amendment 3*. Standard ITU-T P.10/G.100, revision 6.3. ITU-T, July 2008.

123. *Perceptual video quality measurement techniques for digital cable television in the presence of a reduced reference*. Standard ITU-T J.249, revision 1. ITU-T, January 2010.

124. *Objective perceptual multimedia video quality measurement of HDTV for digital cable television in the presence of a full reference*. Standard ITU-T J.341, revision 1. ITU-T, January 2011.

125. *Objective perceptual multimedia video quality measurement of HDTV for digital cable television in the presence of a reduced reference signal*. Standard ITU-T J.342, revision 1. ITU-T, April 2011.

126. *Advanced video coding for generic audiovisual services*. Standard ITU-T H.264, revision 7. ITU-T, April 2012.

127. J.H. Jiang, H.L. Wu, Y. Li, and R.Q. Yu. Three-way data resolution by alternating slice-wise diagonalization (ASD) method. In *Journal of Chemometrics*, 14(1), pp. 15–36, January 2000. ISSN 1099-128X. doi:10.1002/(SICI)1099-128X(200001/02)14:1<15::AID-CEM571>3.0.CO;2-Z.

128. I. Jolliffe. *Principal Component Analysis*. 2nd edition. Springer, New York, 2002. ISBN 978-0-387-95442-4.

129. S. de Jong. SIMPLS: An alternative approach to partial least squares regression. In *Chemometrics and Intelligent Laboratory Systems*, 18(3), pp. 251–263, March 1993. ISSN 0169-7439. doi:10.1016/0169-7439(93)85002-X.

130. S. de Jong. Regression coefficients in multilinear PLS. In *Journal of Chemometrics*, 12(1), pp. 77–81, January 1998. ISSN 1099-128X. doi:10.1002/(SICI)1099-128X(199801/02)12:1<77::AID-CEM496>3.0.CO;2-7.

131. S. de Jong. Regression coefficients in multilinear PLS. In *Journal of Chemometrics*, 12(1), pp. 77–81, January 1998. ISSN 1099-128X. doi:10.1002/(SICI)1099-128X(199801/02)12:1<77::AID-CEM496>3.0.CO;2-7.

132. S. de Jong. Letter from Sijmen de Jong. In *Journal of Chemometrics*, 19(5-7), pp. 270–270, May 2005. ISSN 1099-128X. doi:10.1002/cem.950.

133. S. de Jong and H.A. Kiers. Principal covariates regression: Part I. Theory. In *Chemometrics and Intelligent Laboratory Systems*, 14(1–3), pp. 155–164, April 1992. ISSN 0169-7439. doi:10.1016/0169-7439(92)80100-I.

134. S. de Jong and C.J.F. Ter Braak. Comments on the PLS kernel algorithm. In *Journal of Chemometrics*, 8(2), pp. 169–174, March 1994. ISSN 1099-128X. doi:10.1002/cem.1180080208.

135. Y. Kawayokeita and Y. Horita. NR objective continuous video quality assessment model based on frame quality measure. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 385–388. October 2008. ISBN 978-1-4244-1764-3. ISSN 1522-4880. doi:10.1109/ICIP.2008.4711772.

136. C. Keimel, A.Redl, and K. Diepold. The TUM high definition video data sets. In *Fourth International Workshop on Quality of Multimedia Experience (QoMEX 2012)*, pp. 91–102. July 2012. ISBN 978-1-4673-0725-3. doi:10.1109/QoMEX.2012.6263865.

137. C. Keimel and K. Diepold. On the use of reference monitors in subjective testing for hdtv. In *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, pp. 35–40. June 2010. ISBN 978-1-4244-6959-8. doi:10.1109/QOMEX.2010.5518305.

138. C. Keimel, J. Habigt, and K. Diepold. Challenges in crowd-based video quality assessment. In *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*, pp. 13–18. July 2012. ISBN 978-1-4673-0725-3. doi:10.1109/QoMEX.2012.6263866.

139. 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*, pp. 390–393. October 2010. ISBN 978-1-4244-8111-8. doi:10.1109/MMSP.2010.5662052.

140. C. Keimel, J. Habigt, C. Horch, and K. Diepold. Qualitycrowd – a framework for crowd-based quality evaluation. In *Picture Coding Symposium (PCS), 2012*, pp. 245–248. May 2012. ISBN 978-1-4577-2048-2. doi:10.1109/PCS.2012.6213338.

141. C. Keimel, J. Habigt, C. Horch, and K. Diepold. Video quality evaluation in the cloud. In *Packet Video Workshop (PV), 2012 19th International*, pp. 155–160. May 2012. ISBN 978-1-4673-0299-9. doi:10.1109/PV.2012.6229729.

142. C. Keimel, J. Habigt, M. Klimpke, and K. Diepold. Design of no-reference video quality metrics with multiway partial least squares regression. In *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on*, pp. 49 –54. September 2011. ISBN 978-1-4577-1334-7. doi:10.1109/QoMEX.2011.6065711.

143. C. Keimel, M. Klimpke, J. Habigt, and K. Diepold. No-reference video quality metric for HDTV based on H.264/AVC bitstream features. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 3325–3328. September 2011. ISBN 978-1-4577-1302-6. ISSN 1522-4880. doi:10.1109/ICIP.2011.6116383.

144. C. Keimel, T. Oelbaum, and K. Diepold. Improving the verification process of video quality metrics. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pp. 121–126. July 2009. ISBN 978-1-4244-4370-3. doi: 10.1109/QOMEX.2009.5246966.

145. C. Keimel, T. Oelbaum, and K. Diepold. No-reference video quality evaluation for high-definition video. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pp. 1145–1148. April 2009. ISBN 978-1-4244-2354-5. ISSN 1520-6149. doi:10.1109/ICASSP.2009.4959791.

146. C. Keimel, M. Rothbucher, H. Shen, and K. Diepold. Video is a cube. In *Signal Processing Magazine, IEEE*, 28(6), pp. 41–49, November 2011. ISSN 1053-5888. doi:10.1109/MSP.2011.942468.

147. C. Keimel, A. Redl, and K. Diepold. Influence of viewing experience and stabilization phase in subjective video testing. In F. Gaykema and P.D. Burns (eds.), *Image Quality and System Performance IX*, volume 8293, pp. 829313–1 – 829313–9. January 2012. doi:10.1117/12.907967.

148. C. Keimel, M. Rothbucher, and K. Diepold. Extending video quality metrics to the temporal dimension with 2D-PCR. volume 7867, pp. 786713–1 – 786713–10. SPIE, January 2011. doi:10.1117/12.872406.

149. M.G. Kendall. A new measure of rank correlation. In *Biometrika*, 30(1/2), pp. 81–93, June 1938. ISSN 0006-3444.

150. M.G. Kendall. *Rank Correlation Methods*. 4th edition. Griffin, London, 1970. ISBN 0-85264-199-0.

151. H.A.L. Kiers. Hierarchical relations among three-way methods. In *Psychometrika*, 56(3), pp. 449–470, September 1991. ISSN 0033-3123. doi:10.1007/BF02294485.

152. H.A.L. Kiers. Towards a standardized notation and terminology in multiway analysis. In *Journal of Chemometrics*, 14(3), pp. 105–122, May 2000. ISSN 1099-128X. doi:10.1002/1099-128X(200005/06)14:3<105::AID-CEM582>3.0.CO;2-I.

153. M. Klimpke, C. Keimel, and K. Diepold. *Visuelle Qualitätsmetrik basierend auf der multivariaten Datenanalyse von H.264/AVC Bitstream-Features*. Technical Report, Institute for Data Processing, Technische Universität München, November 2010.

154. T. Kolda. Orthogonal tensor decompositions. In *SIAM Journal on Matrix Analysis and Applications*, 23(1), pp. 243–255, July 2001. ISSN 1095-7162. doi:10.1137/S0895479800368354.

155. T. Kolda and B. Bader. Tensor decompositions and applications. In *SIAM Review*, 51(3), pp. 455–500, September 2009. ISSN 1095-7200. doi:10.1137/07070111X.

156. H. Kong, L. Wang, E.K. Teoh, X. Li, J.G. Wang, and R. Venkateswarlu. Generalized 2D principal component analysis for face image representation and recognition. In *Neural Networks*, 18(5–6), pp. 585–594, August 2005. ISSN 0893-6080. doi:10.1016/j.neunet.2005.06.041.

157. P. Kroonenberg and J. de Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. In *Psychometrika*, 45(1), pp. 69–97, March 1980. ISSN 0033-3123. doi:10.1007/BF02293599.

158. P.M. Kroonenberg. The three-mode company. Online. URL `http://three-mode.leidenuniv.nl/`, accessed on 21.06.2013.

159. P.M. Kroonenberg. *Three-mode principal component analysis. Theory and applications.* DSWO Press, Leiden University, Leiden, 1983.

160. P.M. Kroonenberg. *Applied Mulitway Data Analysis*. John Wiley & Sons, Inc. Hoboken, Hoboken, 2008. ISBN 978-0-470-16497-6.

161. J.B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. In *Linear Algebra and its Applications*, 18(2), pp. 95–138, 1977. ISSN 0024-3795. doi:10.1016/0024-3795(77)90069-6.

162. J. Kruskal. *Multiway Data Analysis*, chapter Rank, decompostion, and uniqueness for 3-way and N-way arrays, pp. 7–18. North-Holland, Amsterdam, 1989. ISBN 0-444-87410-0.

163. W.J. Krzanowski. Cross-validation in principal component analysis. In *Biometrics*, 43(3), pp. 575–584, September 1987. ISSN 0006-341X.

164. Laboratory for Image and Video Engineering (LIVE), The University of Texas at Austin. Live image quality assessment database. Online. URL `http://live.ece.utexas.edu/research/quality/live_video.html`, accessed on 21.06.2013.

165. Laboratory for Image and Video Engineering (LIVE), The University of Texas at Austin. Image & video quality assessment at LIVE. Online, February 2012. URL `http://live.ece.utexas.edu/research/quality/index.htm`, accessed on 28.06.2013.

166. K. Laghari and K. Connelly. Toward total quality of experience: A QoE model in a communication ecosystem. In *Communications Magazine, IEEE*, 50(4), pp. 58–65, April 2012. ISSN 0163-6804. doi:10.1109/MCOM.2012.6178834.

167. E.C. Larson and D.M. Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. In *Journal of Electronic Imaging*, 19(1), pp. 011006–1 – 011006–21, January 2010. ISSN 1017-9909. doi:10.1117/1.3267105.

168. S.C. Larson. The shrinkage of the coefficient of multiple correlation. In *Journal of Educational Psychology*, 22, pp. 45–55, January 1931. ISSN 0022-0663. doi: 10.1037/h0072400.

169. L. de Lathauwer, B. de Moor, and J. Vandewalle. A multilinear singular value decomposition. In *SIAM Journal on Matrix Analysis and Applications*, 21(4), pp. 1253–1278, April 2000. ISSN 1095-7162. doi:10.1137/S0895479896305696.

170. L. de Lathauwer, B. de Moor, and J. Vandewalle. On the best rank-1 and rank-$(R_1, R_2, ..., R_N)$ approximation of higher-order tensors. In *SIAM Journal on Matrix Analysis and Applications*, 21(4), pp. 1324–1342, April 2000. ISSN 1095-7162. doi: 10.1137/S0895479898346995.

171. L. de Lathauwer. *Signal Processing based on Multilinear Algebra*. Ph.D. thesis, Faculty of Engineering, Katholieke Universiteit Leuven, Leuven, September 1997.

172. P. Le Callet, C. Viard-Gaudin, and D. Barba. A convolutional neural network approach for objective video quality assessment. In *Neural Networks, IEEE Transactions on*, 17(5), pp. 1316–1327, September 2006. ISSN 1045-9227. doi:10.1109/TNN.2006.879766.

173. P. Le Callet, S. Möller, and A. Perkis (eds.). *Qualinet White Paper on Definitions of Quality of Experience (2013)*. European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Lausanne, Switzerland, March 2013. Version 1.2.

174. O. Le Meur, A. Ninassi, P. Le Callet, and D. Barba. Overt visual attention for free-viewing and quality assessment tasks: Impact of the regions of interest on a video quality metric. In *Signal Processing: Image Communication*, 25(7), pp. 547–558, August 2010. ISSN 0923-5965. doi:10.1016/j.image.2010.05.006.

175. J.S. Lee, F. De Simone, and T. Ebrahimi. Subjective quality evaluation via paired comparison: Application to scalable video coding. In *Multimedia, IEEE Transactions on*, 13(5), pp. 882 –893, October 2011. ISSN 1520-9210. doi:10.1109/TMM.2011.2157333.

176. J.S. Lee, F. De Simone, N. Ramzan, Z. Zhao, E. Kurutepe, T. Sikora, J. Ostermann, E. Izquierdo, and T. Ebrahimi. Subjective evaluation of scalable video coding for content distribution. In *Proceedings of the international conference on Multimedia*, MM '10, pp. 65–72. ACM, October 2010. ISBN 978-1-60558-933-6. doi:10.1145/1873951.1873981.

177. K. Lee, J. Park, S. Lee, and A. Bovik. Temporal pooling of video quality estimates using perceptual motion models. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 2493–2496. September 2010. ISBN 978-1-4244-7993-1. ISSN 1522-4880. doi:10.1109/ICIP.2010.5652884.

178. 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. In *Consumer Electronics, IEEE Transactions on*, 56(2), pp. 1071–1078, May 2010. ISSN 0098-3063. doi:10.1109/TCE.2010.5506041.

179. A. Leontaris, P. Cosman, and A. Reibman. Quality evaluation of motion-compensated edge artifacts in compressed video. In *Image Processing, IEEE Transactions on*, 16(4), pp. 943 –956, April 2007. ISSN 1057-7149. doi:10.1109/TIP.2007.891778.

180. S. Leurgans and R.T. Ross. Multilinear models: Applications in spectroscopy. In *Statistical Science*, 7(3), pp. 289–310, August 1992. ISSN 0883-4237.

181. W. Lin and C.C.J. Kuo. Perceptual visual quality metrics: A survey. In *Journal of Visual Communication and Image Representation*, 22(4), pp. 297–312, May 2011. ISSN 1047-3203. doi:10.1016/j.jvcir.2011.01.005.

182. F. Lindgren, P. Geladi, and S. Wold. The kernel algorithm for PLS. In *Journal of Chemometrics*, 7(1), pp. 45–59, January 1993. ISSN 1099-128X. doi:10.1002/cem.1180070104.

183. P. Lindh and C. van den Branden Lambrecht. Efficient spatio-temporal decomposition for perceptual processing of video sequences. In *Image Processing, 1996. Proceedings, International Conference on*, volume 3, pp. 331–334. September 1996. ISBN 0-7803-3259-8. doi:10.1109/ICIP.1996.560498.

184. T. Liu, X. Feng, A. Reibman, and Y. Wang. Saliency inspired modeling of packet-loss visibility in decoded video. In *Proceedings of International Workshop on Video Processing and Quality Metrics for Consumer Electronics - VPQM*, pp. 112–116. January 2009.

185. T. Liu, Y. Wang, J. Boyce, Z. Wu, and H. Yang. Subjective quality evaluation of decoded video in the presence of packet losses. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, pp. 1125–1128. April 2007. ISBN 1-4244-0728-1. ISSN 1520-6149. doi:10.1109/ICASSP.2007.366110.

186. T. Liu, Y. Wang, J. Boyce, H. Yang, and Z. Wu. A novel video quality metric for low bit-rate video considering both coding and packet-loss artifacts. In *Selected Topics in Signal Processing, IEEE Journal of*, 3(2), pp. 280 –293, April 2009. ISSN 1932-4553. doi:10.1109/JSTSP.2009.2015069.

187. C.F. van Loan. The ubiquitous kronecker product. In *Journal of Computational and Applied Mathematics*, 123(1–2), pp. 85–100, November 2000. ISSN 0377-0427. doi:10.1016/S0377-0427(00)00393-9.

188. H. Lu, K. Plataniotis, and A. Venetsanopoulos. MPCA: Multilinear principal component analysis of tensor objects. In *Neural Networks, IEEE Transactions on*, 19(1), pp. 18 –39, January 2008. ISSN 1045-9227. doi:10.1109/TNN.2007.901277.

189. J. Lubin. *Vision Models for Target Detection and Recognition*, chapter A visual discrimination model for imaging system design and evaluation, pp. 245–283. World Scientific Publishing, Singapore, 1995. ISBN 978-981-02-2149-2.

190. J. Lubin and D. Fibush. *Sarnoff JND vision model*. Technical Report T1A1.5 Working Group #97-612, ANSI T1 Standards Committee, 1997.

191. F. Lukas and Z. Budrikis. Picture quality prediction based on a visual model. In *Communications, IEEE Transactions on*, 30(7), pp. 1679–1692, July 1982. ISSN 0090-6778. doi:10.1109/TCOM.1982.1095616.

192. R.D. Maesschalck, D. Jouan-Rimbaud, and D. Massart. The mahalanobis distance. In *Chemometrics and Intelligent Laboratory Systems*, 50(1), pp. 1–18, January 2000. ISSN 0169-7439. doi:10.1016/S0169-7439(99)00047-7.

193. P.C. Mahalanobis. On the generalized distance in statistics. In *Proceedings of the national institute of sciences of India*, volume 2, pp. 49–55. April 1936.

194. S. Marčelja. Mathematical description of the responses of simple cortical cells∗. In *Journal of the Optical Society of America A*, 70(11), pp. 1297–1300, Nov 1980. doi:10.1364/JOSA.70.001297.

195. H. Martens and M. Martens. *Multivariate Analysis of Quality*. Wiley & Sons, Chichester, 2001. ISBN 0-471-97428-5.

196. H. Martens and T. Næs. *Near-infrared technology in the agricultural and food industries.*, chapter Multivariate calibration by data compression, pp. 57–87. American Association of Cereal Chemists, Inc., 1987.

197. H. Martens and T. Næs. *Multivariate Calibration*. John Wiley & Sons„ Chichester, January 1993. ISBN 0-417-90979-3.

198. H. Martens. Reliable and relevant modelling of real world data: a personal account of the development of PLS regression. In *Chemometrics and Intelligent Laboratory Systems*, 58(2), pp. 85–95, October 2001. ISSN 0169-7439. doi: 10.1016/S0169-7439(01)00153-8.

199. H. Martens and P. Dardenne. Validation and verification of regression in small data sets. In *Chemometrics and Intelligent Laboratory Systems*, 44(1–2), pp. 99–121, December 1998. ISSN 0169-7439. doi:10.1016/S0169-7439(98)00167-1.

200. H. Martens and M. Martens. Modified jack-knife estimation of parameter uncertainty in bilinear modelling by partial least squares regression (PLSR) . In *Food Quality and Preference*, 11(1-2), pp. 5–16, January 2000. ISSN 0950-3293. doi:10.1016/S0950-3293(99)00039-7.

201. J.P.A. Martins, R.F. Teófilo, and M.M.C. Ferreira. Computational performance and cross-validation error precision of five PLS algorithms using designed and real data sets. In *Journal of Chemometrics*, 24(6), pp. 320–332, June 2010. ISSN 1099-128X. doi:10.1002/cem.1309.

202. P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. A no-reference perceptual blur metric. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pp. III–57 – III–60. 2002. ISBN 0-7803-7622-6. ISSN 1522-4880. doi: 10.1109/ICIP.2002.1038902.

203. M.A. Masry and S.S. Hemami. A metric for continuous quality evaluation of compressed video with severe distortions. In *Signal Processing: Image Communication*, 19(2), pp. 133–146, February 2004. ISSN 0923-5965. doi:10.1016/j.image.2003.08.001.

204. MATLAB. *R2012a*. The MathWorks Inc., Natick, 2012.

205. P. Merkle, A. Smolic, K. Muller, and T. Wiegand. Efficient prediction structures for multiview video coding. In *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(11), pp. 1461–1473, November 2007. ISSN 1051-8215. doi: 10.1109/TCSVT.2007.903665.

206. A. Mittal, A. Moorthy, and A. Bovik. No-reference image quality assessment in the spatial domain. In *Image Processing, IEEE Transactions on*, 21(12), pp. 4695–4708, December 2012. ISSN 1057-7149. doi:10.1109/TIP.2012.2214050.

207. A. Mittal, G. Muralidhar, J. Ghosh, and A. Bovik. Blind image quality assessment without human training using latent quality factors. In *Signal Processing Letters, IEEE*, 19(2), pp. 75–78, February 2012. ISSN 1070-9908. doi:10.1109/LSP.2011.2179293.

208. A. Mittal, R. Soundararajan, and A. Bovik. Making a "completely blind" image quality analyzer. In *Signal Processing Letters, IEEE*, 20(3), pp. 209–212, March 2013. ISSN 1070-9908. doi:10.1109/LSP.2012.2227726.

209. M. Miyahara. Quality assessments for visual service. In *Communications Magazine, IEEE*, 26(10), pp. 51–60, October 1988. ISSN 0163-6804. doi:10.1109/35.7667.

210. M. Miyahara, K. Kotani, and V. Algazi. Objective picture quality scale (PQS) for image coding. In *Communications, IEEE Transactions on*, 46(9), pp. 1215–1226, September 1998. ISSN 0090-6778. doi:10.1109/26.718563.

211. S. Möller. *Quality Engineering - Qualität kommunikationstechnischer Systeme*. Springer, Berlin, 2010. ISBN 978-3-642-11547-9.

212. A. Moorthy and A. Bovik. Efficient video quality assessment along temporal trajectories. In *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(11), pp. 1653–1658, November 2010. ISSN 1051-8215. doi:10.1109/TCSVT.2010.2087470.

213. A. Moorthy and A. Bovik. Blind image quality assessment: From natural scene statistics to perceptual quality. In *Image Processing, IEEE Transactions on*, 20(12), pp. 3350–3364, December 2011. ISSN 1057-7149. doi:10.1109/TIP.2011.2147325.

214. A. Moorthy, L.K. Choi, A. Bovik, and G. De Veciana. Video quality assessment on mobile devices: Subjective, behavioral and objective studies. In *Selected Topics in Signal Processing, IEEE Journal of*, 6(6), pp. 652–671, October 2012. ISSN 1932-4553. doi:10.1109/JSTSP.2012.2212417.

215. A.K. Moorthy and A.C. Bovik. Visual quality assessment algorithms: what does the future hold? In *Multimedia Tools and Applications*, 51(2), pp. 675–696, January 2011. ISSN 1380-7501. doi:10.1007/s11042-010-0640-x.

216. T. Na and M. Kim. A novel no-reference PSNR estimation method with regard to deblocking filtering effect in H.264/AVC bitstreams. In *Circuits and Systems for Video Technology, IEEE Transactions on*, PP(99), pp. 1–1, 2013. ISSN 1051-8215. doi: 10.1109/TCSVT.2013.2255425.

217. M. Naccari, M. Tagliasacchi, and S. Tubaro. No-reference video quality monitoring for H.264/AVC coded video. In *Multimedia, IEEE Transactions on*, 11(5), pp. 932–946, August 2009. ISSN 1520-9210. doi:10.1109/TMM.2009.2021785.

218. M. Narwaria and W. Lin. Objective image quality assessment based on support vector regression. In *Neural Networks, IEEE Transactions on*, 21(3), pp. 515–519, March 2010. ISSN 1045-9227. doi:10.1109/TNN.2010.2040192.

219. M. Nezveda, S. Buchinger, W. Robitza, E. Hotop, P. Hummelbrunner, and H. Hlavacs. Test persons for subjective video quality testing: Experts or non-experts? In *QoEMCS workshop at the EuroITV - 8th European Conference on Interactive TV*. 2010.

220. A. Ninassi, O. Le Meur, P. Le Callet, and D. Barba. Considering temporal variations of spatial visual distortions in video quality assessment. In *Selected Topics in Signal Processing, IEEE Journal of*, 3(2), pp. 253–265, April 2009. ISSN 1932-4553. doi: 10.1109/JSTSP.2009.2014806.

221. T. Næs. Leverage and influence measures for principal component regression. In *Chemometrics and Intelligent Laboratory Systems*, 5(2), pp. 155–168, January 1989. ISSN 0169-7439. doi:10.1016/0169-7439(89)80012-7.

222. T. Næs and I.S. Helland. Relevant components in regression. In *Scandinavian Journal of Statistics*, 20(3), pp. 239–250, 1993. ISSN 0303-6898.

223. T. Næs and B.H. Mevik. Understanding the collinearity problem in regression and discriminant analysis. In *Journal of Chemometrics*, 15(4), pp. 413–426, May 2001. ISSN 1099-128X. doi:10.1002/cem.676.

224. T. Oelbaum, C. Keimel, and K. Diepold. Rule-based no-reference video quality evaluation using additionally coded videos. In *Selected Topics in Signal Processing, IEEE Journal of*, 3(2), pp. 294–303, April 2009. ISSN 1932-4553. doi: 10.1109/JSTSP.2009.2015473.

225. T. Oelbaum, H. Schwarz, M. Wien, and T. Wiegand. Subjective performance evaluation of the SVC extension of H.264/AVC. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 2772–2775. October 2008. ISBN 978-1-4244-1764-3. ISSN 1522-4880. doi:10.1109/ICIP.2008.4712369.

226. T. Oelbaum. *Design and Verification of Video Quality Metrics*. Ph.D. thesis, Department of Electrical Engineering and Information Technology, Technische Universität München, July 2008.

227. T. Oelbaum and K. Diepold. Building a reduced reference video quality metric with very low overhead using multivariate data analysis. In *Proceedings of the 4th International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA'07)*. 2007.

228. T. Oelbaum and K. Diepold. A reduced reference video quality metric for AVC/H. 264. In *European Signal Processing Conference (EUSIPCO), 2007*, pp. 1265–1269. September 2007. ISBN 978-83-921340-2-2.

229. T. Oelbaum, K. Diepold, and W. Zia. A generic method to increase the prediction accuracy of visual quality metrics. In *Picture Coding Symposium (PCS), 2007*. November 2007. ISBN 978-989-8109-05-7.

230. D.W. Osten. Selection of optimal regression models via cross-validation. In *Journal of Chemometrics*, 2(1), pp. 39–48, January 1988. ISSN 1099-128X. doi:10.1002/cem.1180020106.

231. Y.F. Ou, T. Liu, Z. Zhao, Z. Ma, and Y. Wang. Modeling the impact of frame rate on perceptual quality of video. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 689–692. October 2008. ISBN 978-1-4244-1764-3. ISSN 1522-4880. doi:10.1109/ICIP.2008.4711848.

232. Y.F. Ou, Z. Ma, T. Liu, and Y. Wang. Perceptual quality assessment of video considering both frame rate and quantization artifacts. In *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(3), pp. 286–298, March 2011. ISSN 1051-8215. doi:10.1109/TCSVT.2010.2087833.

233. Y.F. Ou, Y. Xue, Z. Ma, and Y. Wang. A perceptual video quality model for mobile platform considering impact of spatial, temporal, and amplitude resolutions. In *IVMSP Workshop, 2011 IEEE 10th*, pp. 117–122. June 2011. ISBN 978-1-4577-1284-5. doi:10.1109/IVMSPW.2011.5970365.

234. Y.F. Ou, Y. Xue, and Y. Wang. A novel quality metric for compressed video considering both frame rate and quantization artifacts. In *Proceedings of International Workshop on Video Processing and Quality Metrics for Consumer Electronics - VPQM*. January 2008.

235. Y.F. Ou, Y. Zhou, and Y. Wang. Perceptual quality of video with frame rate variation: A subjective study. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 2446–2449. March 2010. ISBN 978-1-4244-4296-6. ISSN 1520-6149. doi:10.1109/ICASSP.2010.5496300.

236. P. Paatero. A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. In *Chemometrics and Intelligent Laboratory Systems*, 38(2), pp. 223–242, October 1997. ISSN 0169-7439. doi:10.1016/S0169-7439(97)00031-2.

237. J. Park, K. Seshadrinathan, S. Lee, and A. Bovik. VQPooling: Video quality pooling adaptive to perceptual distortion severity. In *Image Processing, IEEE Transactions on*, 22(2), pp. 610–620, February 2013. ISSN 1057-7149. doi:10.1109/TIP.2012.2219551.

238. J. Park, K. Seshadrinathan, S. Lee, and A. Bovik. Spatio-temporal quality pooling accounting for transient severe impairments and egomotion. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 2509–2512. September 2011. ISBN 978-1-4577-1302-6. ISSN 1522-4880. doi:10.1109/ICIP.2011.6116172.

239. S. Péchard, R. Pépion, and P. Le Callet. Suitable methodology in subjective video quality assessment: a resolution dependent paradigm. In *Proceedings of the Third International Workshop on Image Media Quality and its Applications, IMQA2008*. September 2008.

240. K. Pearson. Note on regression and inheritance in the case of two parents. In *Proceedings of the Royal Society of London*, 58(347-352), pp. 240–242, January 1895. doi:10.1098/rspl.1895.0041.

241. K. Pearson. Mathematical contributions to the theory of evolution. III. regression, heredity, and panmixia. In *Philosophical Transactions of the Royal Society of London A*, 187, pp. 253–318, January 1896. doi:10.1098/rsta.1896.0007.

242. K. Pearson. LIII. on lines and planes of closest fit to systems of points in space. In *Philosophical Magazine Series 6*, 2(11), pp. 559–572, 1901. doi:10.1080/14786440109462720.

243. K. Pearson. Notes on the history of correlation. In *Biometrika*, 13(1), pp. 25–45, October 1920. ISSN 0006-3444.

244. R. Penrose. A generalized inverse for matrices. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 51, pp. 406–413, July 1955. ISSN 1469-8064. doi:10.1017/S0305004100030401.

245. F. Pereira. A triple user characterization model for video adaptation and quality of experience evaluatio. In *Multimedia Signal Processing, 2005 IEEE 7th Workshop on*, pp. 1–4. October-November 2005. ISBN 0-7803-9289-2. doi:10.1109/MMSP.2005.248674.

246. F. Pereira. Sensations, perceptions and emotions towards quality of experience evaluation for consumer electronics video adaptations. In *Proceedings of the First International Workshop on Video Processing and Quality Metrics for Consumer Electronics - VPQM*. January 2005.

247. A. Perkis, S. Munkeby, and O.I. Hillestad. A model for measuring quality of experience. In *Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic*, pp. 198–201. June 2006. ISBN 1-4244-0413-4. doi:10.1109/NORSIG.2006.275209.

248. M. Pinson, L. Janowski, R. Pepion, Q. Huynh-Thu, C. Schmidmer, P. Corriveau, A. Younkin, P. Le Callet, M. Barkowsky, and W. Ingram. The influence of subjects and environment on audiovisual subjective tests: An international study. In *Selected Topics in Signal Processing, IEEE Journal of*, 6(6), pp. 640–651, July 2012. ISSN 1932-4553. doi:10.1109/JSTSP.2012.2215306.

249. M. Pinson and S. Wolf. Comparing subjective video quality testing methodologies. In T. Ebrahimi and T. Sikora (eds.), *SPIE Video Communications and Image Processing Conference*, volume 5150, pp. 8–11. SPIE, June 2003. doi:10.1117/12.509908.

250. M. Pinson and S. Wolf. *The impact of monitor resolution and type on subjective video quality testing*. Technical memorandum TM-04-412, NTIA-ITS, March 2004.

251. M.H. Pinson, K.S. Boyd, J. Hooker, and K. Muntean. How to choose video sequences for video quality assessment. In *Seventh International Workshop on Video Processing and Quality Metrics for Consumer Electronics - VPQM*, pp. 79–85. January 2013.

252. R.M. Pirsig. *Zen and the Art of Motorcycle Maintenance - An inquiry into values*. 2006. Paperback. HarperTorch, New York, 1974. ISBN 978-0-06-058946-2.

253. Y. Pitrey, M. Barkowsky, P. Le Callet, and R. Pepion. Subjective quality assessment of MPEG-4 scalable video coding in a mobile scenario. In *Visual Information Processing (EUVIP), 2010 2nd European Workshop on*, pp. 86–91. July 2010. ISBN 978-1-4244-7288-8. doi:10.1109/EUVIP.2010.5699138.

254. Y. Pitrey, U. Engelke, M. Barkowsky, R. Pepion, and P. Le Callet. Subjective quality of SVC-coded videos with different error-patterns concealed using spatial scalability. In *Visual Information Processing (EUVIP), 2011 3rd European Workshop on*, pp. 180 –185. July 2011. ISBN 978-1-4577-0072-9. doi:10.1109/EuVIP.2011.6045538.

255. Y. Pitrey, M. Barkowsky, P. Le Callet, and R. Pépion. Evaluation of MPEG4-SVC for QoE protection in the context of transmission errors. In A.G. Tescher (ed.), *Applications of Digital Image Processing XXXIII*, volume 7798, p. 77981C. August 2010. doi:10.1117/12.862723.

256. Y. Pitrey, M. Barkowsky, P. Le Callet, and R. Pépion. Subjective Quality Evaluation of H.264 High-Definition Video Coding versus Spatial Up-Scaling and Interlacing. In *QoEMCS workshop at the EuroITV - 8th European Conference on Interactive TV*. Tampere, Finland, June 2010.

257. Y. Pitrey, U. Engelke, M. Barkowsky, R. Pépion, and P. Le Callet. Aligning subjective tests using a low cost common set. In *QoEMCS workshop at the EuroITV - 9th European Conference on Interactive TV*. June 2011.

258. N. Ponomarenko, O. Eremeev, V. Lukin, and K. Egiazarian. Statistical evaluation of no-reference image visual quality metrics. In *Visual Information Processing (EUVIP), 2010 2nd European Workshop on*, pp. 50–54. July 2010. ISBN 978-1-4244-7288-8. doi:10.1109/EUVIP.2010.5699121.

259. N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin. On between-coefficient contrast masking of DCT basis functions. In *Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics - VPQM*. January 2007.

260. E. Poulton. *Bias in Quantifying Judgements*. Lawrence Erlbaum Associates, Hove, 1989. ISBN 0-86377-105-X.

261. M.H. Quenouille. Approximate tests of correlation in time-series. In *Journal of the Royal Statistical Society. Series B (Methodological)*, 11(1), pp. 68–84, 1949. ISSN 0035-9246.

262. R.F. Quick. A vector-magnitude model of contrast detection. In *Biological Cybernetics*, 16(2), pp. 65–67, September 1974. ISSN 0340-1200. doi:10.1007/BF00271628.

263. A. Raake, M.N. Garcia, S. Moller, J. Berger, F. Kling, P. List, J. Johann, and C. Heidemann. T-V-model: Parameter-based prediction of IPTV quality. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 1149–1152. March 2008. ISBN 978-1-4244-1484-0. ISSN 1520-6149. doi: 10.1109/ICASSP.2008.4517818.

264. C. Rao and S. Mitra. *Generalized inverse of matrices and its applications*. John Wiley & Sons, Inc. New York, 1971. ISBN 0-471-70821-6.

265. J. Redi, P. Gastaldo, I. Heynderickx, and R. Zunino. Color distribution information for the reduced-reference assessment of perceived image quality. In *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(12), pp. 1757–1769, December 2010. ISSN 1051-8215. doi:10.1109/TCSVT.2010.2087456.

266. A. Redl, C. Keimel, and K. Diepold. Influence of viewing device and soundtrack in HDTV on subjective video quality. In F. Gaykema and P.D. Burns (eds.), *Image*

Quality and System Performance IX, volume 8293, pp. 829312–1 – 829312–9. SPIE, January 2012. doi:10.1117/12.907015.

267. A. Reibman, R. Bell, and S. Gray. Quality assessment for super-resolution image enhancement. In *Image Processing, 2006 IEEE International Conference on*, pp. 2017–2020. October 2006. ISBN 1-4244-0481-9. ISSN 1522-4880. doi:10.1109/ICIP.2006.312895.

268. T. Richter. SSIM as global quality metric: A differential geometry view. In *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on*, pp. 189–194. September 2011. ISBN 978-1-4577-1334-7. doi:10.1109/QoMEX.2011.6065701.

269. H. de Ridder. Minkowski-metrics as a combination rule for digital-image-coding impairments. In B.E. Rogowitz (ed.), *Human Vision, Visual Processing, and Digital Display III*, pp. 16–26. February 1992. doi:10.1117/12.135953.

270. S. Rimac-Drlje, M. Vranjes, and D. Zagar. Influence of temporal pooling method on the objective video quality evaluation. In *Broadband Multimedia Systems and Broadcasting, 2009. BMSB '09. IEEE International Symposium on*, pp. 1–5. May 2009. ISBN 978-1-4244-2591-4. doi:10.1109/ISBMSB.2009.5133781.

271. J. Riu and R. Bro. Jack-knife technique for outlier detection and estimation of standard errors in PARAFAC models. In *Chemometrics and Intelligent Laboratory Systems*, 65(1), pp. 35–49, January 2003. ISSN 0169-7439. doi:10.1016/S0169-7439(02)00090-4.

272. J.L. Rodgers and W.A. Nicewander. Thirteen ways to look at the correlation coefficient. In *The American Statistician*, 42(1), pp. 59–66, 1988. ISSN 0003-1305. doi:10.1080/00031305.1988.10475524.

273. R. Rosipal. *Chemoinformatics and Advanced Machine Learning Perspectives: Complex Computational Methods and Collaborative Techniques*, chapter Nonlinear Partial Least Squares An Overview, pp. 169–189. IGI Global, Hershey, 2011. ISBN 1-61-520911-5. doi:10.4018/978-1-61520-911-8.ch009.

274. M. Rothbucher, M. Durkovic, H. Shen, and K. Diepold. HRTF customization using multiway array analysis. In *18th European Signal Processing Conference (EUSIPCO), 2010.*, pp. 229–233. August 2010. ISSN 2076-1465.

275. M. Rothbucher, H. Shen, and K. Diepold. Dimensionality reduction in HRTF by using multiway array analysis. In H. Ritter, G. Sagerer, R. Dillmann, and M. Buss (eds.), *Human Centered Robot Systems*, volume 6 of *Cognitive Systems Monographs*, pp. 103–110. Springer, Berlin & Heidelberg, 2009. ISBN 978-3-642-10402-2. doi:10.1007/978-3-642-10403-9_11.

276. D. Ruderman. The statistics of natural images. In *Network: Computation in Neural Systems*, 5(4), pp. 517–548, November 1994. ISSN 0954-898X. doi: 10.1088/0954-898X/5/4/006.

277. M. Saad and A. Bovik. Blind quality assessment of videos using a model of natural scene statistics and motion coherency. In *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*, pp. 332–336. November 2012. ISBN 978-1-4673-5050-1. ISSN 1058-6393. doi: 10.1109/ACSSC.2012.6489018.

278. M. Saad, A. Bovik, and C. Charrier. A DCT statistics-based blind image quality index. In *Signal Processing Letters, IEEE*, 17(6), pp. 583–586, June 2010. ISSN 1070-9908. doi:10.1109/LSP.2010.2045550.

279. M. Saad, A. Bovik, and C. Charrier. Blind image quality assessment: A natural scene statistics approach in the DCT domain. In *Image Processing, IEEE Transactions on*, 21(8), pp. 3339–3352, 2012. ISSN 1057-7149. doi:10.1109/TIP.2012.2191563.

280. R. Schleicher, M.N. Garcia, R. Walter, S. Schlüter, S. Möller, and A. Raake. Where do people look when using combined rating scales? In *16th European Conference on Eye Movements (ECEM)*. August 2011.

281. M.B. Seasholtz and B. Kowalski. The parsimony principle applied to multivariate calibration. In *Analytica Chimica Acta*, 277(2), pp. 165–177, May 1993. ISSN 0003-2670. doi:10.1016/0003-2670(93)80430-S.

282. K. Seshadrinathan and A. Bovik. Motion tuned spatio-temporal quality assessment of natural videos. In *Image Processing, IEEE Transactions on*, 19(2), pp. 335–350, February 2010. ISSN 1057-7149. doi:10.1109/TIP.2009.2034992.

283. K. Seshadrinathan and A. Bovik. Temporal hysteresis model of time varying subjective video quality. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 1153–1156. May 2011. ISBN 978-1-4577-0537-3. ISSN 1520-6149. doi:10.1109/ICASSP.2011.5946613.

284. K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack. Study of subjective and objective quality assessment of video. In *Image Processing, IEEE Transactions on*, 19(6), pp. 1427–1441, June 2010. ISSN 1057-7149. doi:10.1109/TIP.2010.2042111.

285. K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack. A subjective study to evaluate video quality assessment algorithms. In B.E. Rogowitz and T.N. Pappas (eds.), *Human Vision and Electronic Imaging XV*, volume 7527, p. 75270H. January 2010. doi:10.1117/12.845382.

286. C.E. Shannon. A mathematical theory of communication. In *The Bell System Technical Journal,*, 27, pp. 379–423, 623–656, July,October 1948.

287. H. Sheikh and A. Bovik. Image information and visual quality. In *Image Processing, IEEE Transactions on*, 15(2), pp. 430–444, February 2006. ISSN 1057-7149. doi: 10.1109/TIP.2005.859378.

288. H. Sheikh, A. Bovik, and G. De Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. In *Image Processing, IEEE Transactions on*, 14(12), pp. 2117–2128, December 2005. ISSN 1057-7149. doi: 10.1109/TIP.2005.859389.

289. H. Sheikh, M. Sabir, and A. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. In *Image Processing, IEEE Transactions on*, 15(11), pp. 3440–3451, November 2006. ISSN 1057-7149. doi:10.1109/TIP.2006.881959.

290. D. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. 4th edition. Chapman and Hall/CRC, Boca Raton, 2007.

291. A. Shnayderman, A. Gusev, and A. Eskicioglu. An svd-based grayscale image quality measure for local and global assessment. In *Image Processing, IEEE Transactions on*, 15(2), pp. 422–429, February 2006. ISSN 1057-7149. doi:10.1109/TIP.2005.860605.

292. K. Sühring. H.264/AVC JM reference software. Online. URL `http://iphome.hhi.de/suehring/tml/index.htm`, accessed on 21.06.2013.

293. N.D. Sidiropoulos and R. Bro. On the uniqueness of multilinear decomposition of n-way arrays. In *Journal of Chemometrics*, 14(3), pp. 229–239, May 2000. ISSN 1099-128X. doi:10.1002/1099-128X(200005/06)14:3<229::AID-CEM587>3.0.CO;2-N.

294. E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger. Shiftable multiscale transforms. In *Information Theory, IEEE Transactions on*, 38(2), pp. 587–607, March 1992. ISSN 0018-9448. doi:10.1109/18.119725.

295. F. de Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro, and T. Ebrahimi. Subjective assessment of H.264/AVC video sequences transmitted over a noisy channel. In *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*, pp. 204–209. July 2009. ISBN 978-1-4244-4370-3. doi:10.1109/QOMEX.2009.5246952.

296. F. de Simone, M. Tagliasacchi, M. Naccari, S. Tubaro, and T. Ebrahimi. A H.264/AVC video database for the evaluation of quality metrics. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 2430–2433.

March 2010. ISBN 978-1-4244-4296-6. ISSN 1520-6149. doi:10.1109/ICASSP.2010.5496296.

297. M. Slanina, V. Ricny, and R. Forchheimer. A novel metric for H.264/AVC no-reference quality assessment. In *14th International Workshop on Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services.*, pp. 114–117. June 2007. ISBN 978-961-248-029-5. doi:10.1109/IWSSIP.2007.4381166.

298. A.K. Smilde, R. Bro, and P. Geladi. *Multi-way Analysis with Applications in the Chemical Sciences*. John Wiley & Sons, Ltd., Chichester, 2004. ISBN 0-471-98691-7.

299. A.K. Smilde. Comments on multilinear PLS. In *Journal of Chemometrics*, 11(5), pp. 367–377, September 1997. ISSN 1099-128X. doi:10.1002/(SICI)1099-128X(199709/10)11:5<367::AID-CEM481>3.0.CO;2-I.

300. A.K. Smilde and H.A.L. Kiers. Multiway covariates regression models. In *Journal of Chemometrics*, 13(1), pp. 31–48, January 1999. ISSN 1099-128X. doi:10.1002/(SICI)1099-128X(199901/02)13:1<31::AID-CEM528>3.0.CO;2-P.

301. *Critical Viewing Conditions for Evaluation of Color Television Pictures*. Standard SMPTE RP 166, revision 1. SMPTE, January 1995.

302. R. Soundararajan and A. Bovik. Rred indices: Reduced reference entropic differencing for image quality assessment. In *Image Processing, IEEE Transactions on*, 21(2), pp. 517–526, February 2012. ISSN 1057-7149. doi:10.1109/TIP.2011.2166082.

303. R. Soundararajan and A. Bovik. Video quality assessment by reduced reference spatio-temporal entropic differencing. In *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(4), pp. 684–694, April 2013. ISSN 1051-8215. doi:10.1109/TCSVT.2012.2214933.

304. C. Spearman. The proof and measurement of association between two things. In *The American Journal of Psychology*, 15(1), pp. 72–101, January 1904. ISSN 0002-9556.

305. C. Spearman. Demonstration of formulæ for true measurement of correlation. In *The American Journal of Psychology*, 18(2), pp. 161–169, April 1907. ISSN 0002-9556.

306. N. Staelens, D. Deschrijver, E. Vladislavleva, B. Vermeulen, T. Dhaene, and P. Demeester. Constructing a no-reference H.264/AVC bitstream-based video quality metric using genetic programming-based symbolic regression. In *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(99), pp. 1322–1333, August 2013. ISSN 1051-8215. doi:10.1109/TCSVT.2013.2243052.

307. N. Staelens, G. Van Wallendael, K. Crombecq, N. Vercammen, J. De Cock, B. Vermeulen, R. Van De Walle, T. Dhaene, and P. Demeester. No-reference bitstream-based visual quality impairment detection for high definition H.264/AVC encoded video sequences. In *Broadcasting, IEEE Transactions on*, 58(2), pp. 187–199, June 2012. ISSN 0018-9316. doi:10.1109/TBC.2012.2189334.

308. R. Stankiewicz, P. Cholda, and A. Jajszczyk. QoX: What is it really? In *Communications Magazine, IEEE*, 49(4), pp. 148–158, April 2011. ISSN 0163-6804. doi:10.1109/MCOM.2011.5741159.

309. L. Ståhle. Aspects of the analysis of three-way data. In *Chemometrics and Intelligent Laboratory Systems*, 7(1–2), pp. 95–100, December 1989. ISSN 0169-7439. doi:10.1016/0169-7439(89)80114-5. <ce:title>Proceedings of the First Scandinavian Symposium on Chemometrics</ce:title>.

310. P. Stoica and T. Söderström. Partial least squares: A first-order analysis. In *Scandinavian Journal of Statistics*, 25(1), pp. 17–24, March 1998. ISSN 1467-9469. doi:10.1111/1467-9469.00085.

311. M. Stone. Cross-validatory choice and assessment of statistical predictions. In *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2), pp. 111–147, 1974. ISSN 0035-9246.

312. G. Strang. *Introduction to Linear Algebra*. 4th edition. Wellesley-Cambridge Press, Wellesley, 2009. ISBN 978-09802327-14.

313. O. Sugimoto and S. Naito. No reference metric of video coding quality based on parametric analysis of video bitstream. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 3333–3336. September 2011. ISBN 978-1-4577-1302-6. ISSN 1522-4880. doi:10.1109/ICIP.2011.6116385.

314. 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), 16th IEEE International Conference on*, pp. 2237–2240. November 2009. ISBN 978-1-4244-5655-0. ISSN 1522-4880. doi:10.1109/ICIP.2009.5413957.

315. G. Sullivan and T. Wiegand. Video compression - from concepts to the H.264/AVC standard. In *Proceedings of the IEEE*, 93(1), pp. 18 –31, January 2005. ISSN 0018-9219. doi:10.1109/JPROC.2004.839617.

316. E. Svensson. Comparison of the quality of assessments using continuous and discrete ordinal rating scales. In *Biometrical Journal*, 42(4), pp. 417–434, August 2000. ISSN 1521-4036. doi:10.1002/1521-4036(200008)42:4<417::AID-BIMJ417>3.0.CO;2-Z. URL Z.

317. Technische Universität München, Institute for Data Processing. Videolab. Online. URL `http://www.ldv.ei.tum.de/videolab`, accessed on 21.06.2013.

318. P. Teo and D. Heeger. Perceptual image distortion. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 2, pp. 982–986. November 1994. ISBN 0-8186-6952-7. doi:10.1109/ICIP.1994.413502.

319. W.M. Thorburn. The myth of occam's razor. In *Mind*, XXVII(3), pp. 345–353, 1918. ISSN 1460-2113. doi:10.1093/mind/XXVII.3.345.

320. L.L. Thurstone. *Multiple-Factor Analysis*. 7th edition. University of Chicago Press, Chicago, 1965.

321. G. Tomasi and R. Bro. A comparison of algorithms for fitting the parafac model. In *Computational Statistics & Data Analysis*, 50(7), pp. 1700–1734, April 2006. ISSN 0167-9473. doi:10.1016/j.csda.2004.11.013.

322. X. Tong, D.J. Heeger, and C.J. Van den Branden Lambrecht. Video quality evaluation using ST-CIELAB. In B.E. Rogowitz and T.N. Pappas (eds.), *Human Vision and Electronic Imaging IV*, volume 3644, pp. 185–196. January 1999. doi:10.1117/12.348439.

323. J. Trygg and S. Wold. Orthogonal projections to latent structures (O-PLS). In *Journal of Chemometrics*, 16(3), pp. 119–128, March 2002. ISSN 1099-128X. doi:10.1002/cem.695.

324. L.R. Tucker. *Problems in measuring change*, chapter Implications of factor analysis of three-way matrices for measurement of change, pp. 122–137. University of Wisconsin Press, Madison, 1963.

325. L.R. Tucker. *Contributions to mathematical psychology*, chapter The extension of factor analysis to three-dimensional matrices, pp. 110–127. Holt, Rinehart & Winston, New York, 1964.

326. L.R. Tucker. Some mathematical notes on three-mode factor analysis. In *Psychometrika*, 31(3), pp. 279–311, September 1966. ISSN 0033-3123. doi:10.1007/BF02289464.

327. J.W. Tukey. Bias and confidence in not quite large samples. In *The Annals of Mathematical Statistics*, 29(2), p. 614, June 1958. ISSN 0003-4851.

328. Video Quality Experts Group (VQEG). *Final report from the video quality experts group on the validation of objective models of video quality assessment, Phase I*. Technical Report, March 2000.

329. Video Quality Experts Group (VQEG). *Final report from the video quality experts group on the validation of objective models of video quality assessment, Phase II*. Technical Report, August 2003.

330. Video Quality Experts Group (VQEG). *Report on the validation of video quality models for high definition video content*. Technical Report, June 2010.

331. H. van der Voet. Comparing the predictive accuracy of models using a simple randomization test. In *Chemometrics and Intelligent Laboratory Systems*, 25(2), pp. 313–323, November 1994. ISSN 0169-7439. doi:10.1016/0169-7439(94)85050-X.

332. P. Vu, C. Vu, and D. Chandler. A spatiotemporal most-apparent-distortion model for video quality assessment. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 2505–2508. September 2011. ISBN 978-1-4577-1302-6. ISSN 1522-4880. doi:10.1109/ICIP.2011.6116171.

333. B.A. Wandell. *Foundations of Vision*. Sinauer Associates, Sunderland, MA, 1995. ISBN 0-878-93853-2. URL `http://foundationsofvision.stanford.edu/`. Also available online.

334. Z. Wang, E. Simoncelli, and A. Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pp. 1398–1402. November 2003. ISBN 0-7803-8104-1. doi:10.1109/ACSSC.2003.1292216.

335. Z. Wang. IW-SSIM: information content weighted structural similarity index for image quality assessment. Online, January 2011. URL `https://ece.uwaterloo.ca/~z70wang/research/iwssim/`, accessed on 28.06.2013.

336. Z. Wang and A. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. In *Signal Processing Magazine, IEEE*, 26(1), pp. 98–117, January 2009. ISSN 1053-5888. doi:10.1109/MSP.2008.930649.

337. Z. Wang, A. Bovik, and B. Evan. Blind measurement of blocking artifacts in images. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 3, pp. 981–984. September 2000. ISBN 0-7803-6297-7. ISSN 1522-4880. doi:10.1109/ICIP.2000.899622.

338. Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. In *Image Processing, IEEE Transactions on*, 13(4), pp. 600–612, April 2004. ISSN 1057-7149. doi:10.1109/TIP.2003.819861.

339. Z. Wang and A.C. Bovik. *Modern Image Quality Assessment*. Morgan & Claypool Publishers, San Rafael, 2006. ISBN 978-1-59829-022-6.

340. Z. Wang and Q. Li. Information content weighting for perceptual image quality assessment. In *Image Processing, IEEE Transactions on*, 20(5), pp. 1185–1198, May 2011. ISSN 1057-7149. doi:10.1109/TIP.2010.2092435.

341. Z. Wang, L. Lu, and A.C. Bovik. Video quality assessment based on structural distortion measurement. In *Signal Processing: Image Communication*, 19(2), pp. 121–132, February 2004. ISSN 0923-5965. doi:10.1016/S0923-5965(03)00076-6.

342. A. Watson. DCTune: A technique for visual optimization of DCT quantization matrices for individual images. In *Society for Information Display Digest of Technical Papers XXIV*, pp. 946–949. 1993. ISSN 2168-0159.

343. A. Watson, J. Hu, and J. Mcgowan III. DVQ: A digital video quality metric based on human vision. In *Journal of Electronic Imaging*, 10(1), pp. 20–29, January 2001. ISSN 1017-9909. doi:10.1117/1.1329896.

344. A. Watson, Q. Hu, J. McGowan III, and J. Mulligan. Design and performance of a digital video quality metric. In B.E. Rogowitz and T.N. Pappas (eds.), *Human Vision and Electronic Imaging IV*, volume 3644, pp. 168–174. January 1999. doi:10.1117/12.348437.

345. A. Watson and M.A. Sasse. Measuring perceived quality of speech and video in multimedia conferencing applications. In *Proceedings of the sixth ACM international conference on Multimedia*, MULTIMEDIA '98, pp. 55–60. September 1998. ISBN 0-201-30990-4. doi:10.1145/290747.290755.

346. T. Wiegand, L. Noblet, and F. Rovati. Scalable video coding for IPTV services. In *Broadcasting, IEEE Transactions on*, 55(2), pp. 527–538, June 2009. ISSN 0018-9316. doi:10.1109/TBC.2009.2020954.

347. T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. In *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7), pp. 560–576, July 2003. ISSN 1051-8215. doi:10.1109/TCSVT.2003.815165.

348. S. Wiklund, D. Nilsson, L. Eriksson, M. Sjöström, S. Wold, and K. Faber. A randomization test for PLS component selection. In *Journal of Chemometrics*, 21(10-11), pp. 427–439, October 2007. ISSN 1099-128X. doi:10.1002/cem.1086.

349. S. Winkler. A perceptual distortion metric for digital color images. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 3, pp. 399–403. October 1998. ISBN 0-8186-8821-1. doi:10.1109/ICIP.1998.999029.

350. S. Winkler. *Digital Video Image Quality and Perceptual Coding*, chapter Perceptual Video Qualtiy Metrics - A Review, pp. 155–179. CRC Press, Boca Raton, 2006. 0-8427-2777-0.

351. S. Winkler. On the properties of subjective ratings in video quality experiments. In *Quality of Multimedia Experience, 2009. QoMEX 2009. International Workshop on*, pp. 139–144. July 2009. ISBN 978-1-4244-4370-3. doi:10.1109/QOMEX.2009.5246961.

352. S. Winkler. Analysis of public image and video databases for quality assessment. In *Selected Topics in Signal Processing, IEEE Journal of*, 6(6), pp. 616–625, October 2012. ISSN 1932-4553. doi:10.1109/JSTSP.2012.2215007.

353. S. Winkler. Quality metric design: a closer look. In B.E. Rogowitz and T.N. Pappas (eds.), *Human Vision and Electronic Imaging V*, volume 3959, pp. 37–44. June 2000. doi:10.1117/12.387175.

354. S. Winkler. *Digital Video Quality - Vision Models and Metrics*. John Wiley & Sons, Ltd., Chichester, 2005. ISBN 978-0-470-02404-1.

355. H. Wold. *Multivariate Analysis*, chapter Estimation of principal components and related models by iterative least squares, pp. 391–420. Academic Press, New York, 1966.

356. H. Wold. *Quantitative Sociology: International Perspectives on Mathematical and Statistical Model Building*, chapter Path models with latent variables: the NIPALS approach, pp. 307–357. Academic Press, New York, 1975. ISBN 0-12-103950-1.

357. H. Wold. *Systems under Indirect Observation, Part II*, volume 2, chapter Soft modeling: the basic design and some extensions, pp. 1–54. North-Holland, Amsterdam, 1982. ISBN 0-44-486301-X.

358. S. Wold, C. Albano, W. Dunn III, K. Esbensen, S. Hellberg, J. E., and M. Sjöström. *Food Research and Data Analysis.*, chapter Pattern recognition: finding and using patterns in multivariate data, pp. 147–188. Applied Science Publications, London, 1983. ISBN 0-85-334206-7.

359. S. Wold, H. Martens, and H. Wold. The multivariate calibration problem in chemistry solved by the pls method. In B. Kågström and A. Ruhe (eds.), *Matrix Pencils*, volume 973 of *Lecture Notes in Mathematics*, pp. 286–293. Springer Berlin & Heidelberg, 1983. ISBN 978-3-540-11983-8. doi:10.1007/BFb0062108.

360. S. Wold. Cross-validatory estimation of the number of components in factor and principal components models. In *Technometrics*, 20(4), pp. 397–405, 1978. ISSN 0040-1706. doi:10.1080/00401706.1978.10489693.

361. S. Wold. Personal memories of the early PLS development. In *Chemometrics and Intelligent Laboratory Systems*, 58(2), pp. 83–84, October 2001. ISSN 0169-7439. doi:10.1016/S0169-7439(01)00152-6.

362. S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. In *Chemometrics and Intelligent Laboratory Systems*, 2(1–3), pp. 37– 52, August 1987. ISSN 0169-7439. doi:10.1016/0169-7439(87)80084-9.

363. S. Wold, P. Geladi, K. Esbensen, and J. Öhman. Multi-way principal components-and PLS-analysis. In *Journal of Chemometrics*, 1(1), pp. 41–56, January 1987. ISSN 1099-128X. doi:10.1002/cem.1180010107.

364. S. Wold, N. Kettaneh-Wold, and B. Skagerberg. Nonlinear PLS modeling. In *Chemometrics and Intelligent Laboratory Systems*, 7(1-2), pp. 53–65, December 1989. ISSN 0169-7439. doi:10.1016/0169-7439(89)80111-X.

365. S. Wold, M. Sjöström, and L. Eriksson. PLS-regression: a basic tool of chemometrics. In *Chemometrics and Intelligent Laboratory Systems*, 58(2), pp. 109–130, 2001. ISSN 0169-7439. doi:10.1016/S0169-7439(01)00155-1.

366. S. Wolf. Measuring the end-to-end performance of digital video systems. In *Broadcasting, IEEE Transactions on*, 43(3), pp. 320–328, September 1997. ISSN 0018-9316. doi:10.1109/11.632940.

367. S. Wolf and M. Pinson. *Video Quality Measurement Techniques*. Report 02-329, NTIA-ITS, June 2002.

368. H.R. Wu and K.R. Rao (eds.). *Digital Video Image Quality and Perceptual Coding*. CRC Press, Boca Raton, 2006. ISBN 0-8247-2777-0.

369. T. Yamada, S. Yachida, Y. Senda, and M. Serizawa. Accurate video-quality estimation without video decoding. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 2426–2429. March 2010. ISBN 978-1-4244-4296-6. ISSN 1520-6149. doi:10.1109/ICASSP.2010.5496285.

370. F. Yang, S. Wan, Q. Xie, and H.R. Wu. No-reference quality assessment for networked video via primary analysis of bit stream. In *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(11), pp. 1544–1554, November 2010. ISSN 1051-8215. doi:10.1109/TCSVT.2010.2087433.

371. J. Yang, D. Zhang, A. Frangi, and J.Y. Yang. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1), pp. 131–137, January 2004. ISSN 0162-8828. doi:10.1109/TPAMI.2004.1261097.

372. J. You, J. Korhonen, and A. Perkis. Spatial and temporal pooling of image quality metrics for perceptual video quality assessment on packet loss streams. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 1002–1005. March 2010. ISBN 978-1-4244-4296-6. ISSN 1520-6149. doi:10.1109/ICASSP.2010.5495313.

373. J. You, J. Korhonen, A. Perkis, and T. Ebrahimi. Balancing attended and global stimuli in perceived video quality assessment. In *Multimedia, IEEE Transactions on*, 13(6), pp. 1269–1285, December 2011. ISSN 1520-9210. doi:10.1109/TMM.2011.2172591.

374. J. You, U. Reiter, M.M. Hannuksela, M. Gabbouj, and A. Perkis. Perceptual-based quality assessment for audio-visual services: A survey. In *Signal Processing: Image Communication*, 25(7), pp. 482–501, August 2010. ISSN 0923-5965. doi:10.1016/j.image.2010.02.002.

375. Q. Zhao, C. Caiafa, D. Mandic, Z. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki. Higher-order partial least squares (hopls): A generalized multi-linear regression method. In *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(7), pp. 1660–1673, 2013. ISSN 0162-8828. doi:10.1109/TPAMI.2012.254.

376. S. Zieliński, F. Rumsey, and S. Bech. On some biases encountered in modern audio quality listening tests-a review. In *Journal of the Audio Engineering Society*, 56(6), pp. 427–451, June 2008. ISSN 1549-4950.

Appendices

A. Data Analysis

Appendix A provides additional information about and references for some of the mathematical methods used in this thesis in the context of two-way and multi-way data analysis.

A.1. Algebra of two-way arrays

In general, a basic knowledge of linear algebra as provided in common textbooks e.g. in Strang [312] is sufficient for understanding the concepts presented in this thesis. But in the context of unfolded multi-way arrays, some matrix products are used that are not necessarily widely known and are therefore briefly presented in this section. The definitions are based on Smilde et al. [298], and Cichocki et al. [45].

A.1.1. Outer product

The outer product of two vectors $\mathbf{a} \in \mathbb{R}^{I\times 1}$ and $\mathbf{b} \in \mathbb{R}^{J\times 1}$ yields a matrix $\mathbf{C} \in \mathbb{R}^{I\times J}$ as

$$\mathbf{C} = \mathbf{a} \circ \mathbf{b} = \mathbf{a}\mathbf{b}^{\top}. \tag{A.1.1}$$

For convenience the operator symbol $\circ$ is omitted in this thesis if it is obvious from the context and type of fibres that the outer product is used i.e. if the result of the product of two one-way arrays is a two-way array.

A.1.2. Inner product

The inner or scalar product of two matrices $\mathbf{A} \in \mathbb{R}^{I\times J}$ and $\mathbf{B} \in \mathbb{R}^{I\times J}$ yielding a scalar c is a denoted as $\langle \mathbf{A}, \mathbf{B} \rangle \in \mathbb{R}$ and defined as

$$c = \langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i=1}^{I} \sum_{j=1}^{J} a_{ij} b_{ij}. \tag{A.1.2}$$

Note, that $\mathbf{A}$ and $\mathbf{B}$ need to be of equal size. This product is sometimes also called *Frobenius inner product*, as it is used in the *Frobenius norm* for matrices, defined as

$$\|\mathbf{A}\|_F = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle} = \sqrt{\sum_{i=1}^{I} \sum_{j=1}^{J} a_{ij} b_{ij}}. \tag{A.1.3}$$

A.1.3. Kronecker product

The Kronecker product of two matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ is denoted as $\mathbf{A} \otimes \mathbf{B} \in \mathbb{R}^{IK \times JL}$ and defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix}. \tag{A.1.4}$$

With $\mathbf{a}$ and $\mathbf{b}$ as the column vectors of $\mathbf{A}$ and $\mathbf{B}$ the Kronecker product can also be written as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_1 \otimes \mathbf{b}_2 & \mathbf{a}_1 \otimes \mathbf{b}_3 & \cdots & \mathbf{a}_J \otimes \mathbf{b}_{L-1} & \mathbf{a}_J \otimes \mathbf{b}_L \end{bmatrix}. \tag{A.1.5}$$

Some properties of the Kronecker product are:

$$a \otimes \mathbf{A} = a\mathbf{A} = \mathbf{A}a = \mathbf{A} \otimes a \tag{A.1.6a}$$

$$(\mathbf{A} \otimes \mathbf{B})^\top = \mathbf{A}^\top \otimes \mathbf{B}^\top \tag{A.1.6b}$$

$$\mathbf{A} \otimes \mathbf{B} \otimes \mathbf{C} = (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) \tag{A.1.6c}$$

$$(\mathbf{A} + \mathbf{B}) \otimes (\mathbf{C} + \mathbf{D}) = \mathbf{A} \otimes \mathbf{C} + \mathbf{A} \otimes \mathbf{D} + \mathbf{B} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{D} \tag{A.1.6d}$$

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}. \tag{A.1.6e}$$

With the Kronecker product, the outer product between two column vectors $\mathbf{a}$ and $\mathbf{b}$ can be written as

$$\mathbf{a}\mathbf{b}^\top = \mathbf{a} \otimes \mathbf{b}^\top = \mathbf{b}^\top \otimes \mathbf{a}. \tag{A.1.7}$$

For a comprehensive list of the Kronecker product's properties I also refer to van Loan [187].

A.1.4. Hadamard product

The Hadamard or element wise product of two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{I \times J}$ is denoted as $\mathbf{A} \circledast \mathbf{B} \in \mathbb{R}^{I \times J}$ and defined as

$$\mathbf{A} \circledast \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix}. \tag{A.1.8}$$

Note, that $\mathbf{A}$ and $\mathbf{B}$ need to be of equal size. Some properties of the Kronecker product are:

$$\mathbf{A} \circledast \mathbf{B} = \mathbf{B} \circledast \mathbf{A} \tag{A.1.9a}$$

$$(\mathbf{A} \circledast \mathbf{B})^\top = \mathbf{A}^\top \circledast \mathbf{B}^\top \tag{A.1.9b}$$

$$(\mathbf{A} \circledast \mathbf{B}) \circledast \mathbf{C} = \mathbf{A} \circledast (\mathbf{B}) \circledast \mathbf{C} \tag{A.1.9c}$$

$$(\mathbf{A} + \mathbf{B}) \circledast (\mathbf{C} + \mathbf{D}) = \mathbf{A} \circledast \mathbf{C} + \mathbf{A} \circledast \mathbf{D} + \mathbf{B} \circledast \mathbf{C} + \mathbf{B} \circledast \mathbf{D}. \tag{A.1.9d}$$

For a square matrix $\mathbf{A} \in \mathbb{R}^{I \times J}$ also

$$\mathbf{A} \circledast \mathbf{I} = \operatorname{diag}(a_{11}, \cdots, a_{II}), \tag{A.1.10}$$

holds, where $\mathbf{I}$ is the identity matrix.

A.1.5. Khatri-Rao product

For matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ that are partitioned in the equal number of M partitions as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \cdots \mathbf{A}_M \end{bmatrix} \tag{A.1.11a}$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \cdots \mathbf{B}_M \end{bmatrix}, \tag{A.1.11b}$$

the Khatri-Rao product $\mathbf{A} \odot \mathbf{B} \in \mathbb{R}^{IJ \times KL}$ of $\mathbf{A}$ and $\mathbf{B}$ is defined as

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{A}_1 \otimes \mathbf{B}_1 & \cdots & \mathbf{A}_M \otimes \mathbf{B}_M \end{bmatrix}, \tag{A.1.12}$$

and some properties of the Khatri-Rao product are:

$$\mathbf{A} \odot \mathbf{B} \neq \mathbf{B} \odot \mathbf{A} \tag{A.1.13a}$$

$$(\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C} = \mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}) \tag{A.1.13b}$$

$$(\mathbf{A} + \mathbf{B}) \odot (\mathbf{C} + \mathbf{D}) = \mathbf{A} \odot \mathbf{C} + \mathbf{A} \odot \mathbf{D} + \mathbf{B} \odot \mathbf{C} + \mathbf{B} \odot \mathbf{D}. \tag{A.1.13c}$$

If $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times J}$ have the same number of columns, the special case that $L = J$, where the matrices are partitioned in their J columns, the Khatri-Rao product can be written as

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_J \otimes \mathbf{b}_J \end{bmatrix}, \tag{A.1.14}$$

with $\mathbf{A} \odot \mathbf{B} \in \mathbb{R}^{IK \times J}$ and where $\mathbf{a}_j$ and $\mathbf{b}_j$ are the column vectors of $\mathbf{A}$ and $\mathbf{B}$, respectively. The Khatri-Rao product is therefore also called column Kronecker product. In this special case, also the following additional property holds

$$(\mathbf{A} \odot \mathbf{B})^\top (\mathbf{A} \odot \mathbf{B}) = (\mathbf{A}^\top \mathbf{A}) \circledast (\mathbf{B}^\top \mathbf{B}). \tag{A.1.15}$$

Additionally, the Khatri-Rao product between two vectors $\mathbf{a}$ and $\mathbf{b}$ is equivalent to the Kronecker product between $\mathbf{a}$ and $\mathbf{b}$

$$\mathbf{a} \odot \mathbf{b} = \mathbf{a} \otimes \mathbf{b}. \tag{A.1.16}$$

A.1.6. Vec-operator

The vec-operator stacks all columns vectors of a matrix underneath each other, resulting in one column vector. For a matrix $\mathbf{A} \in \mathbb{R}^{I \times J}$ with columns $\mathbf{a}_j \in \mathbb{R}^{I \times 1}$, $\operatorname{vec} \mathbf{A} \in \mathbb{R}^{IJ \times 1}$ is therefore defined as

$$\operatorname{vec} \mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_J \end{bmatrix}. \tag{A.1.17}$$

Some properties of the vec-operator are:

$$\operatorname{vec}(\mathbf{a}^\top) = \operatorname{vec} \mathbf{a} = \mathbf{a} \tag{A.1.18a}$$

$$\operatorname{vec}(\mathbf{a}\mathbf{b}^\top) = \mathbf{b} \otimes \mathbf{a} \tag{A.1.18b}$$

$$\operatorname{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A}) \operatorname{vec} \mathbf{B}. \tag{A.1.18c}$$

A.2. Algebra of multi-way arrays

This section briefly introduces some of the algebra operations used in this thesis for handling multi-way arrays based on Kolda and Bader [155], and Cichocki et al. [45]. Although the definitions are only given for the three-way case, all operations can be extended to n-way arrays with $n > 3$ and I refer to [155] and [45] for more information.

A.2.1. Addition and subtraction

For two three-way arrays $\underline{\mathbf{A}} \in \mathbb{R}^{I \times J \times K}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{I \times J \times K}$, where all modes have the same dimensionality, the addition for the sum

$$\underline{\mathbf{C}} = \underline{\mathbf{A}} + \underline{\mathbf{B}} \tag{A.2.1}$$

is defined for each element c_{ijk} of $\underline{\mathbf{C}}$ as

$$c_{ijk} = a_{ijk} + b_{ijk}, \tag{A.2.2}$$

and the subtraction $\underline{\mathbf{C}} = \underline{\mathbf{A}} - \underline{\mathbf{B}}$ is defined analogously.

A.2.2. Outer product

The outer product of two three-way arrays $\underline{\mathbf{A}} \in \mathbb{R}^{I \times J \times K}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{L \times M \times N}$ is denoted as $\underline{\mathbf{A}} \circ \underline{\mathbf{B}} \in \mathbb{R}^{I \times J \times K \times L \times M \times N}$ and for the product

$$\underline{\mathbf{C}} = \underline{\mathbf{A}} \circ \underline{\mathbf{B}} \tag{A.2.3}$$

each element c_{ijklmn} is defined as

$$c_{ijklmn} = a_{ijk} b_{lmn}. \tag{A.2.4}$$

Extending the outer product of two vectors from (A.1.1) to three vectors $\mathbf{a} \in \mathbb{R}^{I\times 1}$, $\mathbf{b} \in \mathbb{R}^{J\times 1}$ and $\mathbf{c} \in \mathbb{R}^{K\times 1}$ yields a three-way array $\mathbf{D} \in \mathbb{R}^{I\times J\times K}$ as

$$\underline{\mathbf{D}} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}, \tag{A.2.5}$$

where each element d_{ijk} is defined as

$$d_{ijk} = a_i b_j c_k. \tag{A.2.6}$$

For convenience the operator symbol $\circ$ is again omitted in this thesis if it is obvious from the context and type of fibres that the outer product is used.

A.2.3. Inner product

The inner or scalar product of two three-way arrays $\underline{\mathbf{A}} \in \mathbb{R}^{I\times J\times K}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{I\times J\times K}$ yielding a scalar c is a denoted as $\langle \underline{\mathbf{A}}, \underline{\mathbf{B}} \rangle \in \mathbb{R}$ and defined as

$$c = \langle \underline{\mathbf{A}}, \underline{\mathbf{B}} \rangle = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} a_{ijk} b_{ijk}. \tag{A.2.7}$$

Note, that $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ need to be of equal size. Similar to the inner product for two-way arrays, this product is used in the Frobenius norm for multi-way arrays, defined as

$$\|\underline{\mathbf{A}}\|_F = \sqrt{\langle \underline{\mathbf{A}}, \underline{\mathbf{A}} \rangle} = \sqrt{\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} a_{ijk} b_{ijk}}. \tag{A.2.8}$$

In this thesis, the Forbenius norm of a three-way array $\underline{\mathbf{A}}$ is usually denoted as $\|\underline{\mathbf{A}}\|$, if it is obvious from the context that the norm is applied to a three-way array.

A.2.4. Mode-n product

The mode-n product allows the multiplication of a three-way array with a two-way array along the n-th mode of the three-way array and is denoted as $\times_n$. Note, that the dimensionality of the n-th mode of the three-way array and the second mode (columns) of the two-way arrays must be the same, as the mode-n fibre of the three-way array is multiplied by the two-way array.

For a three-way array $\underline{\mathbf{G}} \in \mathbb{R}^{I\times J\times K}$ and the two-way arrays $\mathbf{A} \in \mathbb{R}^{P\times I}$, $\mathbf{B} \in \mathbb{R}^{Q\times J}$ and $\mathbf{C} \in \mathbb{R}^{R\times K}$, the mode-1 , mode-2 and mode-3 product are denoted as $\underline{\mathbf{G}} \times_1 \mathbf{A}$, $\underline{\mathbf{G}} \times_2 \mathbf{B}$ and $\underline{\mathbf{G}} \times_3 \mathbf{C}$, resulting in the following three-way arrays

$$\underline{\mathbf{D}} = \underline{\mathbf{G}} \times_1 \mathbf{A} \in \mathbb{R}^{P\times J\times K} \tag{A.2.9a}$$

$$\underline{\mathbf{E}} = \underline{\mathbf{G}} \times_2 \mathbf{B} \in \mathbb{R}^{I\times Q\times K} \tag{A.2.9b}$$

and

$$\underline{\mathbf{F}} = \underline{\mathbf{G}} \times_3 \mathbf{C} \in \mathbb{R}^{I \times J \times R}, \tag{A.2.9c}$$

respectively, where each element is defined as

$$d_{pjk} = \sum_{i=1}^{I} g_{ijk} a_{pi}, \tag{A.2.10a}$$

$$e_{iqk} = \sum_{j=1}^{J} g_{ijk} b_{qj}, \tag{A.2.10b}$$

and

$$f_{ijr} = \sum_{k=1}^{K} g_{ijk} c_{rk}. \tag{A.2.10c}$$

The mode-n product can also be expressed using the unfolded representation of the three way array. For the mode-1 product of $\underline{\mathbf{G}}$ and $\mathbf{A}$, for example,

$$\underline{\mathbf{D}} = \underline{\mathbf{G}} \times_1 \mathbf{A} \tag{A.2.11a}$$

is equivalent to

$$\mathbf{D}_{P \times JK} = \mathbf{A}\mathbf{G}_{I \times JK}, \tag{A.2.11b}$$

where $\mathbf{D}_{P \times JK}$ and $\mathbf{G}_{I \times JK}$ are the mode-1 unfolded three-way arrays $\underline{\mathbf{D}}$ and $\underline{\mathbf{G}}$, respectively. The mode-2 and mode-3 product can be expressed similarly with the mode-2 and mode-3 unfolding of the three-way arrays. The mode-n product can be applied successively along different modes and is commutative, so that for $n \neq m$

$$(\underline{\mathbf{G}} \times_n \mathbf{A}) \times_m \mathbf{B} = (\underline{\mathbf{G}} \times_m \mathbf{B}) \times_n \mathbf{A} = \underline{\mathbf{G}} \times_n \mathbf{A} \times_m \mathbf{B}, \tag{A.2.12}$$

and for appropriate $\mathbf{A}$ and $\mathbf{B}$ the consecutive multiplication along the same mode can also be written as

$$(\underline{\mathbf{G}} \times_n \mathbf{A}) \times_n \mathbf{B} = \underline{\mathbf{G}} \times_n (\mathbf{B}\mathbf{A}). \tag{A.2.13}$$

The mode-n product can also be used to perform a multiplication between a three-way array and one-way array along the n-th mode. With the three-way array three-way array $\underline{\mathbf{G}} \in \mathbb{R}^{I \times J \times K}$ and the one-way arrays $\mathbf{a} \in \mathbb{R}^{I \times 1}$, $\mathbf{b} \in \mathbb{R}^{J \times 1}$ and $\mathbf{c} \in \mathbb{R}^{K \times 1}$, the mode-1 , mode-2 and mode-3 product results in the following two-way arrays

$$\mathbf{D} = \underline{\mathbf{G}} \times_1 \mathbf{a} \in \mathbb{R}^{J \times K} \tag{A.2.14a}$$

$$\mathbf{E} = \underline{\mathbf{G}} \times_2 \mathbf{b} \in \mathbb{R}^{I \times K} \tag{A.2.14b}$$

and

$$\mathbf{F} = \underline{\mathbf{G}} \times_3 \mathbf{c} \in \mathbb{R}^{I \times J}, \tag{A.2.14c}$$

respectively, where each element is defined as

$$d_{jk} = \sum_{i=1}^{I} g_{ijk} a_i, \tag{A.2.15a}$$

$$e_{ik} = \sum_{j=1}^{J} g_{ijk} b_j, \tag{A.2.15b}$$

and

$$f_{ij} = \sum_{k=1}^{K} g_{ijk} c_k. \tag{A.2.15c}$$

As before for the multiplication with two-way arrays, the mode-n product with one-way arrays can also be expressed using the unfolded representation of the three way array. For the mode-1 product of $\underline{\mathbf{G}}$ and $\mathbf{a}$, for example,

$$\mathbf{D} = \underline{\mathbf{G}} \times_1 \mathbf{a} \tag{A.2.16a}$$

is equivalent to

$$\mathbf{D} = \mathbf{a}^\top \mathbf{G}_{I \times JK}, \tag{A.2.16b}$$

where $\mathbf{G}_{I \times JK}$ is the mode-1 unfolded three-way array $\underline{\mathbf{D}}$. The mode-2 and mode-3 product can be expressed similarly with the mode-2 and mode-3 unfolding of the three-way arrays.

Note, that instead of three-way arrays, the result of the mode-n product between a three-way array and a one-way array is a two-way array. The multiplication can also be performed on more than one mode: if the three-way array is multiplied in two modes by a one-way array, the result is a one-way array and if multiplied in all three-modes by a one-way array, the result is scalar or zero-way array. As the order of the multi-way array changes by consecutive multiplication with one-way arrays in each multiplication, the multiplication with one-way arrays is not commutative:

$$\underline{\mathbf{G}} \times_m \mathbf{a} \times_n \mathbf{b} = (\underline{\mathbf{G}} \times_m \mathbf{a}) \times_{n-1} \mathbf{b} = (\underline{\mathbf{G}} \times_n \mathbf{b}) \times_m \mathbf{a}. \tag{A.2.17}$$

A.2.5. Diagonal multi-way array

A *diagonal multi-way array* describes a multi-way array where all off-superdiagonal elements are zero. It is an extension of the concept of diagonal two-way arrays to multi-way

arrays and the *superdiagonal* is also defined as an extension of the diagonal in two-way arrays. For a three-way array $\underline{\mathbf{A}} \in \mathbb{R}^{I \times J \times K}$ this can be expressed for each element a_{ijk} as

$$a_{ijk} = \begin{cases} 0 & \text{if } i \neq j \neq k \\ \neq 0 & \text{if } i = j = k. \end{cases} \tag{A.2.18}$$

If the multi-way array is also a hypercube, it is also called *superdiagonal multi-way array*. For a three-way array it follows that if $I = J = K$ and therefore $\underline{\mathbf{A}} \in \mathbb{R}^{I \times I \times I}$ is a cube, $\underline{\mathbf{A}}$ is a superdiagonal three-way array [160].

A.2.6. Identity multi-way array

The *identity multi-way array* is a special case of a diagonal multi-way array where all superdiagonal elements are one and is denoted as $\underline{\mathbf{I}}$. It is a straightforward extension of the identity matrix $\mathbf{I}$ for two-way arrays to multi-way arrays. For three-way arrays and the corresponding identity three-way array $\underline{\mathbf{I}} \in \mathbb{R}^{I \times J \times K}$ with $I = J = K$, this can be expressed as for each element i_{ijk} as

$$i_{ijk} = \begin{cases} 0 & \text{if } i \neq j \neq k \\ 1 & \text{if } i = j = k. \end{cases} \tag{A.2.19}$$

The properties of $\underline{\mathbf{I}}$ are similar as for the identity matrix $\mathbf{I}$ for two-way arrays, only extended to multi-way arrays.

A.2.7. Rank

The *rank* of a three-way array $\underline{\mathbf{A}} \in \mathbb{R}^{I \times J \times K}$ is similar defined to the rank of matrices as the smallest number of rank-one three-way arrays represented by triads that generate $\underline{\mathbf{A}}$ as their sum [161]. Hence if $\underline{\mathbf{A}}$ can be exactly decomposed in R components as

$$\underline{\mathbf{A}} = \sum_{r=1}^{R} \mathbf{b}_r \mathbf{c}_r \mathbf{d}_r, \tag{A.2.20}$$

then the rank of $\underline{\mathbf{A}}$ is R, denoted as

$$\operatorname{rank} \underline{\mathbf{A}} = R. \tag{A.2.21}$$

A.2.8. *N*-rank

The *n-rank* of a three-way array $\underline{\mathbf{A}} \in \mathbb{R}^{I \times J \times K}$ is defined as the (column-)rank of the mode-n unfolding of $\underline{\mathbf{A}}$ [162]. It represents the dimension of the vector space spanned by the mode-n fibres.The 1-rank of $\underline{\mathbf{A}}$ along the first mode is given by

$$\operatorname{rank}_1 \underline{\mathbf{A}} = \operatorname{rank} \underline{\mathbf{A}}_{I \times JK}, \tag{A.2.22}$$

where $\underline{\mathbf{A}}_{I\times JK}$ is the mode-1 unfolding of $\underline{\mathbf{A}}$. The n-rank for the other modes is determined similarly. With R, S and T as the 1-rank, 2-rank and 3-rank of $\underline{\mathbf{A}}$, respectively, we can describe $\underline{\mathbf{A}}$ as a rank-(R, S, T) three-way array [155].

A.2.9. Formal notation of unfolding

Unfolding as discussed in Section 4.2.1 can also be described in a more formal notation, but although *conceptually simple, the formal notation is clunky* [155] for multi-way arrays in general. For three-way arrays, however, the resulting notation is still rather pleasant.

Assuming a three-way array $\underline{\mathbf{A}} \in \mathbb{R}^{I\times J\times K}$, the mode-1 unfolded representation is given as $\mathbf{A}_{I\times JK} \in \mathbb{R}^{I\times JK}$ and introducing the index $p = jk$ for convenience, a element of $\mathbf{A}_{I\times JK}$ is then denoted as a_{ip}. For an element a_{ijk} of $\underline{\mathbf{A}}$, the corresponding element a_{ip} in $\mathbf{A}_{I\times JK}$ is then identified by the index p as

$$p = j + (k - 1)J, \tag{A.2.23}$$

where j and k are the mode-2 and mode-3 indices of the element in the three-way array, respectively, and J is the dimensionality of the second mode in $\underline{\mathbf{A}}$. Note, that the index i of the first mode remains unchanged. For the mode-2 unfolding and mode-3 unfolding of $\underline{\mathbf{A}}$, the new index p for the corresponding elements in $\mathbf{A}_{I\times JK}$ is given by

$$p = i + (k - 1)I \tag{A.2.24a}$$

and

$$p = i + (j - 1)I, \tag{A.2.24b}$$

respectively. For more information about unfolding, in particular for multi-way arrays, I refer to Cichocki et al. [45] and Kolda and Bader [155].

A.3. Multi-way component models

This section provides some further details on multi-way component models and their properties that are helpful to understand some of the issues discussed in Section 6.2.

A.3.1. Tucker3 and the projection of three-way arrays

As briefly discussed in the properties of the PARAFAC model in Section 6.2, the projection of three-way arrays on subspaces represented by the component or loading matrices of a PARAFAC decomposition equals a Tucker3 model and therefore although PARAFAC provides the best low-rank approximation of a three-way array, PARAFAC does not provide the best subspace approximation of a three-way array with respect to the variation explained

by the subspace. According to Bro et al. [33], this can be explained as following: considering the two-way PCA on a two-way array $\mathbf{D} \in \mathbb{R}^{I \times J}$ resulting in scores $\mathbf{T} \in \mathbb{R}^{I \times R}$ and loadings $\mathbf{P} \in \mathbb{R}^{J \times R}$ for R principal components, the approximation of $\mathbf{D}$ is then given by

$$\mathbf{D} = \mathbf{T}\mathbf{P}^\top + \mathbf{E} \tag{A.3.1}$$

and the projection $\mathbf{D}^{Proj}$ of $\mathbf{D}$ on the subspaces defined by $\mathbf{T}$ and $\mathbf{P}$ can be written as

$$\mathbf{D}^{Proj} = \mathbf{T}\mathbf{T}^+\mathbf{D}(\mathbf{P}\mathbf{P}^+)^\top, \tag{A.3.2}$$

with

$$\mathbf{T}\mathbf{T}^+\mathbf{D}(\mathbf{P}\mathbf{P}^+)^\top = \mathbf{T}\mathbf{P}^\top, \tag{A.3.3}$$

which can be verified by plucking (A.3.1) into (A.3.3). As the orthogonal projectors $\mathbf{T}\mathbf{T}^+$ and $\mathbf{P}\mathbf{P}^+$ are orthogonal to $\mathbf{E}$, $\mathbf{E}$ is therefore annihilated in $\mathbf{T}\mathbf{T}^+\mathbf{D}(\mathbf{P}\mathbf{P}^+)^\top$, whereas $\mathbf{T}\mathbf{P}^\top$ remains unaffected.

Extending this to a three-way array $\underline{\mathbf{D}} \in \mathbb{R}^{I \times J \times K}$, the approximation of $\underline{\mathbf{D}}$ with the PARAFAC model and the corresponding loadings $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ is given for the mode-1 unfolding of $\underline{\mathbf{D}}$ by

$$\mathbf{D}_{I \times JK} = \mathbf{A}(\mathbf{B} \odot \mathbf{C})^\top + \mathbf{E}_{I \times JK}. \tag{A.3.4}$$

The projection $\underline{\mathbf{D}}^{Proj}$ of $\underline{\mathbf{D}}$ on the subspaces defined by $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ can be gained by consecutively using the orthogonal projectors $\mathbf{A}\mathbf{A}^+ \in \mathbb{R}^{I \times I}$, $\mathbf{B}\mathbf{B}^+ \in \mathbb{R}^{J \times J}$ and $\mathbf{C}\mathbf{C}^+ \in \mathbb{R}^{K \times K}$ on $\underline{\mathbf{D}}$ in the first, second and third mode, respectively. Using the unfolded representations of $\underline{\mathbf{D}}$, firstly the mode-1 unfolding $\mathbf{D}_{I \times JK}$ is projected onto $\mathbf{A}\mathbf{A}^+$ resulting in a new three-way array $\mathbf{F}_{I \times JK} = \mathbf{A}\mathbf{A}^+\mathbf{D}_{I \times JK}$ with $\underline{\mathbf{F}} \in \mathbb{R}^{I \times J \times K}$, which is in turn unfolded along the second mode and projected onto the second mode resulting in $\mathbf{H}_{J \times IK} = \mathbf{B}\mathbf{B}^+\mathbf{F}_{J \times IK}$ with $\underline{\mathbf{G}} \in \mathbb{R}^{J \times I \times K}$. Lastly, $\underline{\mathbf{G}}$ is unfolded along the third mode and projected on the third mode resulting in $\mathbf{G}_{K \times JI} = \mathbf{C}\mathbf{C}^+\mathbf{G}_{K \times JI}$ with $\underline{\mathbf{H}} \in \mathbb{R}^{K \times J \times I}$ and rearranging $\underline{\mathbf{H}}$ appropriately we arrive at $\underline{\mathbf{D}}^{Proj} \in \mathbb{R}^{I \times J \times K}$. The overall projection onto all three modes can then be written analogously to (A.3.2) as

$$\mathbf{D}^{Proj}_{I \times JK} = \mathbf{A}\mathbf{A}^+\mathbf{D}_{I \times JK}(\mathbf{C}\mathbf{C}^+ \otimes \mathbf{B}\mathbf{B}^+)^\top. \tag{A.3.5}$$

Obviously, this is not a PARAFAC model, but rather a Tucker3 model with the loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ and using the mode-1 unfolding representation of the Tucker3 model, the Tucker3 model can be written as

$$\mathbf{D}_{I \times JK} = \mathbf{A}\mathbf{G}_{R \times RR}(\mathbf{C} \otimes \mathbf{B})^\top. \tag{A.3.6}$$

Inserting the core array $\underline{\mathbf{G}} \in \mathbb{R}^{R \times R \times R}$ given by

$$\mathbf{G}_{R \times RR} = \mathbf{A}^+\mathbf{D}_{I \times JK}(\mathbf{C}^+ \otimes \mathbf{B}^+)^\top \tag{A.3.7}$$

into (A.3.6), we arrive at the projection given in (A.3.5), clearly showing that the projection on the subspaces defined by the loading matrices of the PARAFAC model is represented by a Tucker3 model. Hence, the PARAFAC decomposition is the best low-rank approximation, but the best subspace approximation is the Tucker3 decomposition.

A.3.2. Calculating scores of new samples for Tucker3 models

For an existing Tucker3 model of a three-way array $\underline{\mathbf{D}} \in \mathbb{R}^{I \times J \times K}$ with the core array $\underline{\mathbf{G}} \in \mathbb{R}^{R \times S \times U}$ and the loading matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times S}$ and $\mathbf{C} \in \mathbb{R}^{K \times U}$ the scores $\mathbf{a} \in \mathbb{R}^{R}$ of a new sample represented by the two-way array $\mathbf{D}_{new} \in \mathbb{R}^{J \times K}$ are determined by

$$\mathbf{a} = ((\mathbf{C} \otimes \mathbf{B})\mathbf{G}_{R \times US}^{\top})^{+} \operatorname{vec} \mathbf{D}_{new}, \tag{A.3.8}$$

where we assume that the first mode is the mode representing the samples and therefore the score are represented by the components of the first mode [298].

A.3.3. Calculating scores of new samples for PARAFAC models

For an existing PARAFAC model of a three-way array $\underline{\mathbf{D}} \in \mathbb{R}^{I \times J \times K}$ with the loading matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$ and $\mathbf{C} \in \mathbb{R}^{K \times R}$ the scores $\mathbf{a} \in \mathbb{R}^{R}$ of a new sample represented by the two-way array $\mathbf{D}_{new} \in \mathbb{R}^{J \times K}$ are determined by

$$\mathbf{a} = (\mathbf{C} \odot \mathbf{B})^{+} \operatorname{vec} \mathbf{D}_{new}, \tag{A.3.9}$$

where we assume that the first mode is the mode representing the samples and therfore the score are represented by the components of the first mode [298].

A.4. Model building considerations

This section provides additional information on some aspects discussed in Chapter 7 that should be considered during the model building process.

A.4.1. Cross validated squared correlation coefficient

Instead of the mean square error of prediction (MSEP) discussed in the context of cross validation in Section 7.1, Anderssen et al. [3] suggest to express the prediction error also as the cross validated squared correlation coefficient q^2 that takes not only the MSEP into account, but also the sample variance s^2 of the prediction error. The cross validated squared correlation coefficient q^2 is then given by

$$q^2 = 1 - \frac{MSEP}{s^2} = 1 - \frac{\sum_{n=1}^{N}(y_n - \widehat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \overline{y})^2}, \tag{A.4.1}$$

where y_n is the visual quality of the n-th video sequence, $\widehat{y}_n$ the visual quality prediction of the n-th sequence gained with a model where the n-the sequence has been left out of the calibration set and $\overline{y}$ is the average visual quality over all sequences. As we can see from (A.4.1), the smaller the prediction error is compared to the sample variance of the visual quality of all sequences, the higher the cross validated squared correlation will be. Thus

q^2 represents the fraction of the variation explained during the cross validation. In contrast to the PRESS plot with the MSEP in Section 7.1, the aim is therefore now to achieve a maximum for the criterion represented by q^2.

A.4.2. Leverage in multiple linear regression

Although the interpretation of the leverage as the squared Mahalanobis distance in Section 7.4 allows for a more general application of the leverage concept, it is helpful for the understanding of the leverage concept to review its original definition for regression models by Hoaglin and Welsch [88]. Considering a data two-way array $\mathbf{A} \in \mathbb{R}^{I \times J}$, a vector $\mathbf{c} \in \mathbb{R}^{I \times 1}$ and the regression weight vector $\mathbf{b} \in \mathbb{R}^{J \times 1}$, the leverage vector $\mathbf{h} \in \mathbb{R}^{J \times 1}$ for the regression problem $\mathbf{c} = \mathbf{Ab} + \mathbf{e}$ can be expressed as

$$\mathbf{h} = \operatorname{diag}\left[\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-\mathbf{1}}\mathbf{A}^\top\right], \tag{A.4.2}$$

where the i-th element in $\mathbf{h}$ describes the leverage of the i-th object and $0 \leq h_i \leq 1$. By rewriting the solution to the regression problem with $\mathbf{b} = \mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-\mathbf{1}}\mathbf{A}^\top$ as [298]

$$\widehat{\mathbf{c}} = \mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-\mathbf{1}}\mathbf{A}^\top\mathbf{c}, \tag{A.4.3}$$

the interpretation of the leverage becomes clear: if the i-th leverage h_i corresponding to the i-th diagonal element of $\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-\mathbf{1}}\mathbf{A}^\top$ is zero, it can be shown that all elements in the i-th row and column of $\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-\mathbf{1}}\mathbf{A}^\top$ are zero. Hence the i-th sample represented by c_i has no influence on the regression at all. The opposite is true if the i-th leverage h_i is one: then the (i, i)-th element of $\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-\mathbf{1}}\mathbf{A}^\top$ is one and all other elements of $\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-\mathbf{1}}\mathbf{A}^\top$ are zero and thus the prediction $\widehat{c}_i$ of c_i is only influenced by c_i. Hence leverage provides a useful measure of influence of the different samples [298].

B. Video Quality

Appendix B provides additional details on selected topics of video quality, the performance metrics and also further results of the performance comparison.

B.1. Subjective testing methods

Table B.1 on the following page provides an overview of most commonly used subjective testing methods. General information about some of the methods in Table B.1 is provided in Section 2.5.2 and for more detailed information, I refer to the references provided in Table B.1 .

B.2. Definitions of selected video quality metrics

Most video quality metrics can not be concisely described in equations, but rather consist of one or more different algorithms. Two exceptions are the *peak-signal-to-noise ratio (PSNR)* and the *structural-similarity-index (SSIM)*. Although both are image quality metrics, they are often also used for video by considering each frame as an image, followed by temporally pooling the image quality of each frame over all frames Both can be calculated with comparably simple equations that are presented in this section.

B.2.1. PSNR

The peak-signal-to-noise ratio (PSNR) is a commonly used pixel based metric used in image and video processing. The advantage of PSNR compared to the mean squared error (MSE) lies mostly in the comparison of images with different dynamic ranges and as it is a logarithmic representation of a ratio in dB, attenuation and gain can easily be determined by addition. For a grey-scale, discrete image or frame of a video sequence $\mathbf{S} \in \mathbb{N}^{U \times V}$ with $U \times V$ pixel and 8 bit per pixel, the processed or distorted version of $\mathbf{S}$ is denoted as $\widehat{\mathbf{S}}$ and the MSE between $\mathbf{S}$, and $\widehat{\mathbf{S}}$ is then given by

$$MSE = \frac{1}{UV} \sum_{u=1}^{U} \sum_{u=1}^{U} (s_{uv} - \widehat{s}_{uv})^2, \tag{B.2.1}$$

where the element s_{uv} and $\widehat{s}_{uv}$ describe the intensity of the (u, v)-th pixel with $u = 1, 2, \ldots, U$ and $v = 1, 2, \ldots, V$. The corresponding PSNR for an 8 bit image resulting in

Table B.1.: Overview subjective testing methods

Method	Methodology	Scale			Labels	Standard
		type	quality	range		
SSIS	single stimulus	discrete	impairment	5	categorical & numerical	ITU-R BT.500 [109]
SSMM			quality	11		ITU-R BT.500 [109]
ACR		discrete[a]	quality	5,9,11		ITU-T P.910 [121]
ACR-HR			quality	5,9,11		ITU-T P.910 [121]
SSCQE	continuous evaluation	continuous	quality	-	categorical	ITU-R BT.500 [109]
DSIS	double stimulus	discrete	impairment	5	categorical & numerical	ITU-R BT.500 [109]
DCR			impairment	5		ITU-T P.910 [121]
DSUR			quality	11		Baroncini [12]
DSCQS		continuous	quality	-	categorical	ITU-R BT.500 [109]
SDSCE	continuous evaluation	continuous	quality	-	categorical	ITU-R BT.500 [109]
SAMVIQ	interactive	continuous	quality	0-100	categorical & numerical	ITU-R BT.1788 [104]
PC	pair comparison	-	-	-	-	ITU-T P.910 [121]

[a] Continuous scale is optional in ITU-T P.910 [121]

a dynamic range of $L = 2^8 - 1$ can then be expressed as

$$PSNR = 10 \log_{10} \frac{L^2}{MSE}. \tag{B.2.2}$$

Usually for simplicity only the luma component of the images or video frames version is used in the calculation of the PSNR, resulting in the so-called *luma PSNR*. Depending on the context, sometimes the PSNR is also determined for the chroma components.

Considering that we have T frames in our video sequence, the PSNR for the complete video sequence is often expressed as the average of the individual $PSNR_t$ of each frame as

$$PSNR = \frac{1}{T} \sum_{t=1}^{T} PSNR_t. \tag{B.2.3}$$

Note, however, that (B.2.3) does *not* correspond to the arithmetic mean of the individual frames' $PSNR_t$ over all frames T, but rather to the geometric mean of the individual frames' $PSNR_t$ over all frames T. If the true average PSNR of a video sequence is desired, the averaging needs to performed for the MSE before calculating the PSNR, resulting in the global MSE_T of the complete video sequence as

$$MSE_T = \frac{1}{TUV} \sum_{t=1}^{T} \sum_{u=1}^{U} \sum_{u=1}^{U} (s_{tuv} - \widehat{s}_{tuv})^2, \tag{B.2.4}$$

where the element s_{tuv} and $\widehat{s}_{tuv}$ describe the intensity of the (u, v)-th pixel in the t-th frame. Using (B.2.4) in (B.2.3) provides then the true average PSNR over all T frames.

B.2.2. SSIM

The structural similarity index (SSIM) by Wang et al. [338] is currently one of the most ubiquitous image and by extension video quality metrics. This is mainly due to its straightforward calculation that is presented in this section.

The main idea behind the SSIM is the comparison between the undistorted reference and the distorted version of the reference as a function of luminance, contrast and structure of both. Assuming a grey-scale, discrete image or frame of a video sequence $\mathbf{S} \in \mathbb{N}^{U \times V}$ with $U \times V$ pixel and denoting the processed or distorted version of $\mathbf{S}$ as $\widehat{\mathbf{S}}$, this can be written using the notation from [338] as

$$SSIM = l(\mathbf{S}, \widehat{\mathbf{S}}) c(\mathbf{S}, \widehat{\mathbf{S}}) s(\mathbf{S}, \widehat{\mathbf{S}}), \tag{B.2.5}$$

where $l(\mathbf{S}, \widehat{\mathbf{S}})$ $c(\mathbf{S}, \widehat{\mathbf{S}})$ and $s(\mathbf{S}, \widehat{\mathbf{S}})$ describe the luminance, contrast and structure comparison of $\mathbf{S}$ and $\widehat{\mathbf{S}}$, respectively. For convenience, I introduce the notation $\mathbf{S} = \mathbf{A}$ and $\widehat{\mathbf{S}} = \mathbf{B}$. Then $\mu_{\mathbf{A}}$ and $\mu_{\mathbf{B}}$ are then the sample mean intensity[1] of pixels in $\mathbf{A}$ and $\mathbf{B}$, respectively, as

[1] Following the notation in Wang et al. [338], I denote the sample mean and sample variation as μ and σ, respectively, even tough this notation is usually only used for the population mean and variation.

given by

$$\mu_{\mathbf{A}} = \frac{1}{UV}\sum_{u=1}^{U}\sum_{v=1}^{V} a_{uv} \tag{B.2.6a}$$

and

$$\mu_{\mathbf{B}} = \frac{1}{UV}\sum_{u=1}^{U}\sum_{v=1}^{V} b_{uv}, \tag{B.2.6b}$$

and $\sigma_{\mathbf{A}}$ and $\sigma_{\mathbf{B}}$ is the sample variance of the intensity in A and B, respectively, as given by

$$\sigma_{\mathbf{A}} = \sqrt{\frac{1}{UV-1}\sum_{u=1}^{U}\sum_{v=1}^{V}(a_{uv}-\mu_{\mathbf{A}})^2} \tag{B.2.7a}$$

and

$$\sigma_{\mathbf{B}} = \sqrt{\frac{1}{UV-1}\sum_{u=1}^{U}\sum_{v=1}^{V}(b_{uv}-\mu_{\mathbf{B}})^2}. \tag{B.2.7b}$$

Lastly, we define the sample cross correlation $\sigma_{\mathbf{AB}}$ as

$$\sigma_{\mathbf{AB}} = \frac{1}{UV-1}\sum_{u=1}^{U}\sum_{v=1}^{V}(a_{uv}-\mu_{\mathbf{A}})(b_{uv}-\mu_{\mathbf{B}}). \tag{B.2.8}$$

The SSIM between A and B is then given by

$$SSIM = \left(\frac{2\mu_{\mathbf{A}}\mu_{\mathbf{B}}+C_1}{\mu_{\mathbf{A}}^2+\mu_{\mathbf{B}}^2+C_1}\right)\left(\frac{2\sigma_{\mathbf{A}}\sigma_{\mathbf{B}}+C_2}{\sigma_{\mathbf{A}}^2+\sigma_{\mathbf{B}}^2+C_2}\right)\left(\frac{\sigma_{\mathbf{AB}}+C_3}{\sigma_{\mathbf{A}}\sigma_{\mathbf{B}}+C_3}\right). \tag{B.2.9}$$

where the first term represents the luminance comparison $l(\mathbf{A}, \widehat{\mathbf{B}})$, the second term the contrast comparison $c(\mathbf{A}, \widehat{\mathbf{B}})$ and the third term the structure comparison $c(\mathbf{A}, \widehat{\mathbf{B}})$ from (B.2.5). Setting $C_3 = C_2/2$ we arrive at the commonly used form of the SSIM as

$$SSIM = \frac{(2\mu_{\mathbf{A}}\mu_{\mathbf{B}}+C_1)(2\sigma_{\mathbf{AB}}+C2)}{(\mu_{\mathbf{A}}^2+\mu_{\mathbf{B}}^2+C_1)(\sigma_{\mathbf{A}}^2+\sigma_{\mathbf{B}}^2+C_2)}. \tag{B.2.10}$$

C_1, C_2 and C_3 represent small constants that stabilise each term, but can usually be set to $C_1 = C_2 = C_3 = 0$ [336]. For more details on the SSIM and especially a justification of the comparison function, I refer to [336, 338].

B.3. Definitions of performance metrics

In this section the performance metrics used in the performance comparison in Chapter 9 are discussed in detail.

B.3.1. Pearson correlation

The correlation coefficient most commonly used in the research on video quality metrics is the *Pearson product-moment correlation coefficient* [240, 241] or *Pearson's r*, often just called *Pearson correlation*. It provides a measure of how strong the linear relationship between two variables, in our case $\widehat{\mathbf{y}}$ and $\mathbf{y}$, is. The (sample) Pearson correlation coefficient r_p for N samples i.e. video sequences can be determined as

$$r_p = \frac{\sum_{n=1}^{N}(y_n - \overline{y})(\widehat{y}_n - \overline{\widehat{y}})}{\sqrt{\sum_{n=1}^{N}(y_n - \overline{y})^2}\sqrt{\sum_{n=1}^{N}(\widehat{y}_n - \overline{\widehat{y}})^2}}, \tag{B.3.1}$$

where y_n and $\widehat{y}_n$ are the visual quality and visual quality estimation of the n-th sequence, and $\overline{y}$ and $\overline{\widehat{y}}$ are the average over all N samples. The correlation coefficient r_p can take values between, -1.0 and 1.0, i.e. $-1.0 \leq r_p \leq 1.0$. The upper limit, 1.0 indicates that all pairs of y_n and $\widehat{y}_n$ lie on one line and $\widehat{y}_n$ increases with increasing y_n, whereas the lower limit of -1.0 indicates that even though both y_n and $\widehat{y}_n$ lie on one line, $\widehat{y}_n$ decreases with increasing y_n. Note that it does not necessarily prove that there is really a linear relationship between two variables as illustrated by Anscombe [6] with the so-called Anscombe's quartet.

In order to do determine the 95% confidence interval for r_p, we firstly apply the Fisher transformation F [63] to r_p by

$$z_p = F(r_p) = \operatorname{arctanh}(r_p) = \frac{1}{2}\ln\left(\frac{1 + r_p}{1 - r_p}\right), \tag{B.3.2}$$

to obtain the corresponding z-score z_p. This is necessary, as especially for the desired high values of r_p, the Pearson correlation coefficient is negatively skewed due to its limitation to the interval $[-1.0, 1.0]$. We can then determine the upper and lower limit of the 95% confidence interval of z_p, $z_{p,u}$ and $z_{p,l}$, by

$$z_{p,u} = z_p + \frac{1.96}{\sqrt{N-3}} \tag{B.3.3a}$$

and

$$z_{p,l} = z_p - \frac{1.96}{\sqrt{N-3}}, \tag{B.3.3b}$$

where N represents the number of samples, 1.96 is the corresponding standard score for the 95% interval for the normally distributed z_p and the term

$$s_z = \frac{1}{\sqrt{N-3}} \tag{B.3.4}$$

is the standard error s_z of z_p, where we assumed that $N > 30$. For $N \leq 30$, Student's t distribution with $N-1$ degrees of freedom must be used for determining the values corresponding to the (two-tailed) 95% confidence intervals. If the 99% confidence interval is required, the factor 1.96 needs to be replaced by the corresponding standard score of 2.58. By applying the inverse Fisher transformation

$$r_{p,u} = F^{-1}(z_{p,u}) = \tanh(z_{p,u}) = \frac{e^{2z_{p,u}} - 1}{e^{2z_{p,u}} + 1} \tag{B.3.5a}$$

and

$$r_{p,l} = F^{-1}(z_{p,l}) = \tanh(z_{p,l}) = \frac{e^{2z_{p,l}} - 1}{e^{2z_{p,l}} + 1}, \tag{B.3.5b}$$

we then obtain the upper and lower limits of the 95% confidence interval for r_p, $r_{p,u}$ and $r_{p,l}$. Hence, the interval $[r_{p,u}, r_{p,l}]$ contains r_p with a confidence level of 95% or alternatively expressed, the probability that the interval $[r_{p,u}, r_{p,l}]$ contains the true value of r_p is 95%.

Additionally, the statistical significance of the difference between the correlation coefficients of two different models can be evaluated with the z-scores gained by the Fisher transformation. The null hypothesis H_0 is that there is no significant difference between the two correlation coefficients and the alternative hypothesis H_1 that there is a significant difference between the two correlation coefficients. Assuming the prediction abilities of two models for the same data with N samples are represented by z_{p_1} and z_{p_2} , corresponding to the Fisher transformed correlation coefficients r_{p_1} and r_{p_2} of model 1 and model 2, respectively, the statistic

$$z = \frac{z_{p_1} - z_{p_2}}{\sqrt{s_z^2 + s_z^2}} \tag{B.3.6}$$

allows us to determine the p-value of H_0 using either the (two-tailed) normal distribution for $N > 30$ or Student's t-distribution for $N \leq 30$. If $p < 0.05$, we reject the null hypothesis H_0 and the correlation coefficients r_{p_1} and r_{p_2} are therefore significantly different at the 0.05 level or, expressed differently, there is a likelihood of less than 5% that r_{p_1} and r_{p_2} are different only due to chance.

For an in-depth discussion of the properties of the Pearson correlation coefficient, but also the other correlations coefficients used in this thesis, I refer to the comprehensive overview provided by Sheskin [290]. For an overview of the history of the Pearson product-moment correlation coefficient, I refer to [272] and [243].

B.3.2. Spearman rank order correlation

Another correlation coefficient frequently used in the research on video quality metrics is the *Spearman rank order correlation coefficent* or *Spearman's* ρ [304, 305], often just called *Spearman rank correlation*. It is similar to the Pearson correlation, but instead of the variables themselves uses their corresponding rank. It provides a measure of the strength of the monotonicity between the two variables. Let γ_n and $\widehat{\gamma}_n$ be the rank of the visual quality y_n and the visual quality estimation $\widehat{y}_n$ of the n-th sequence. The Spearman correlation r_s for N samples i.e. video sequences can then be determined by

$$r_s = \frac{\sum_{n=1}^{N}(\gamma_n - \overline{\gamma})(\widehat{\gamma}_n - \overline{\widehat{\gamma}})}{\sqrt{\sum_{n=1}^{N}(\gamma_n - \overline{\gamma})^2}\sqrt{\sum_{n=1}^{N}(\widehat{\gamma}_n - \overline{\widehat{\gamma}})^2}}, \tag{B.3.7}$$

where $\overline{\gamma}$ and $\overline{\widehat{\gamma}}$ are the average over all N samples. The correlation coefficient r_s can take values between, -1.0 and 1.0, i.e. $-1.0 \leq r_s \leq 1.0$. Similar to the Pearson correlation, the upper limit, 1.0 indicates that a perfect increasing monotonic relationship and -1.0 that a perfect decreasing monotonic relationship between all y_n and $\widehat{y}_n$ exists. The former means that for any given sample n all $\widehat{y}_n$ and y_n have the same rank, the latter means that the ranking of $\widehat{y}_n$ and y_n is inverse to each other.

The 95% confidence interval can be determined similar to the Pearson correlation by using the Fisher transformation F in (B.3.2)

$$z_s = F(r_s), \tag{B.3.8}$$

in order to obtain the corresponding z-score z_s. Instead of the standard error in (B.3.4) the term [62]

$$s_z = \sqrt{\frac{1.06}{N-3}} \tag{B.3.9}$$

for the standard error s_z of z_s should be used that accounts for the differences in the distribution of the Spearman rank coefficient. The upper and lower limits of the 95% confidence interval and the statistical significance of the difference between two correlation coefficients can then be calculated similar to the Pearson correlation coefficient and only the standard error s_z in (B.3.3), (B.3.5) and (B.3.6) needs to be replaced with the standard error of the Fisher transformed Spearman correlation coefficient from (B.3.9).

B.3.3. Kendall rank order correlation

Similar to the Spearman rank correlation, the *Kendall rank correlation* or *Kendall's* τ [149, 150], uses the rank of variables to determine a measure of their relationship. But unlike the Spearman rank correlation that can be considered an extension of the Pearson correlation to ranks, the Kendall rank order correlation follows a different approach. As before, let γ_n

and $\widehat{\gamma}_n$ be the rank of the visual quality y_n and the visual quality estimation $\widehat{y}_n$ of the n-th sequence. The Kendall rank correlation r_k for N samples can then be determined by

$$r_k = \frac{N_C - N_D}{\frac{1}{2}N(N-1)} = \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\operatorname{sgn}(\gamma_i - \gamma_j)\operatorname{sgn}(\widehat{\gamma}_i - \widehat{\gamma}_j)}{\frac{1}{2}N(N-1)}, \tag{B.3.10}$$

where N_C is the number of concordant pairs, N_D the number of discordant pairs and the denominator represents the number of all possible pairs. The concordant pairs are those pairs y_n and $\widehat{y}_n$, for which if $\gamma_i > \gamma_j$, also $\widehat{\gamma}_i > \widehat{\gamma}_j$ holds true. The discordant pairs are those pairs y_n and $\widehat{y}_n$ for which this condition does not hold true. The correlation coefficient r_k can take values between -1.0 and 1.0, i.e. $-1.0 \leq r_k \leq 1.0$, where 1.0 indicates that there is a perfect match between both rankings and -1.0 that there is a perfect inverse match between the ranking of all y_n and $\widehat{y}_n$. It can also be interpreted as an estimation of the probability that $\widehat{\mathbf{y}}$ is correctly ordered: if it has the same order as $\mathbf{y}$, r_k equals 1 but the more discordant pairs we get i.e. with increasing mismatch in the ranking between $\widehat{\mathbf{y}}$ and $\mathbf{y}$, the more r_k approaches 0. Note also, that whereas the Spearman rank correlation penalises large rank errors stronger, e.g. if y_n has the best rating and $\widehat{y}_n$ has the worst rating, the Kendall rank correlation does not weigh the magnitude of the error, but just considers how many rank order errors occurred. Equation (B.3.10), also called τ_A, assumes that there are no ties within the ranking. If ties within the ranking occur i.e. two sequences i and j in $\widehat{\mathbf{y}}$ or $\mathbf{y}$ have exactly the same value, modified versions of Kendall's τ, τ_B and τ_C are available that take possible ties and also the structure of the data into account [150].

The z-score z_k of p_k with a standard normal distribution can be determined for $N > 10$ samples as [290]

$$z_k = \frac{r_k}{s_z} = 3r_k\sqrt{\frac{N(N-1)}{2(2N+5)}}, \tag{B.3.11}$$

where s_z is the standard error given by

$$s_z = \frac{1}{3}\sqrt{\frac{2(2N+5)}{N(N-1)}}. \tag{B.3.12}$$

Note, that unlike for the Pearson correlation or Spearman rank order correlation no additional transformation is necessary, as the Kendall correlation coefficient for $N > 10$ samples already well approximates the normal distribution up to scaling with the standard error. We can then determine the upper and lower limit of the 95% confidence interval of r_k, $r_{k,u}$ and $r_{k,l}$, by

$$r_{k,u} = r_k + 1.96s_z \tag{B.3.13a}$$

and

$$r_{k,l} = r_k - 1.96s_z \tag{B.3.13b}$$

where 1.96 is the corresponding standard score for the 95% interval for the normally distributed z_k and we used the relationship from (B.3.11) between z_k and r_k to express the critical value of 1.96 for the confidence interval in terms of the non-normalised correlation coefficient. For $N \leq 30$, Student's t distribution with $N-1$ degrees of freedom must be used for determining the values corresponding to the (two-tailed) 95% confidence intervals. If the 99% confidence interval is required, the factor 1.96 needs to be replaced by the corresponding standard score of 2.58.

The statistical significance of the difference between two correlation coefficients can then be calculated similar to the Pearson correlation coefficient and only the standard error s_z in (B.3.3), (B.3.5) and (B.3.6) needs to be replaced with the standard error of the Kendall correlation coefficient from (B.3.12).

B.3.4. Root-mean-square error (RMSE)

The *root-mean-square error (RMSE)* is another commonly used performance metric in the field of video quality metrics. It provides a measure of the absolute error between the visual quality $\mathbf{y}$ and its prediction $\widehat{\mathbf{y}}$. The RMSE for N samples can easily be determined by

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \widehat{y}_n)^2}, \tag{B.3.14}$$

where y_n and $\widehat{y}_n$ are once again the visual quality and visual quality estimation of the n-th sequence. In this thesis, I follow the definition of the MSE and consequently the RSME by Martens and Næs [197] as a measurement of the quality of the prediction $\widehat{\mathbf{y}}$ or, using the notation by Martens and Næs, the *root mean square error of prediction* (*RMSEP*). Note, that the Video Quality Experts Group (VQEG) [330] and ITU-T P.1401 [110] define the RMSE differently as the standard error of the prediction and its unbiased estimator is then given by

$$RMSE_{VQEG} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(y_n - \widehat{y}_n)^2}. \tag{B.3.15}$$

One can argue, however, that for large N the bias introduced by using N in (B.3.14) instead of $N-1$ in (B.3.15) is negligible. The main reason that (B.3.15) is preferred by VQEG is that VQEG usually performs a curve fitting of the metrics' prediction results and therefore the degrees of freedom d are reduced further, necessitating the use of $N-d$ instead of $N-1$ in the denominator of (B.3.15), as depending on the mapping function and resulting d the bias introduced by omitting this correction of the degrees of freedom would become significant. According to [110], the RMSE is approximately χ^2 distributed and the upper and lower limit of the 95% confidence interval of the RMSE, $RMSE_u$ and $RMSE_l$ are therefore given by

$$RMSE_u = \frac{RMSE\sqrt{N-1}}{\sqrt{\chi^4_{0.025}(N-1)}} \tag{B.3.16a}$$

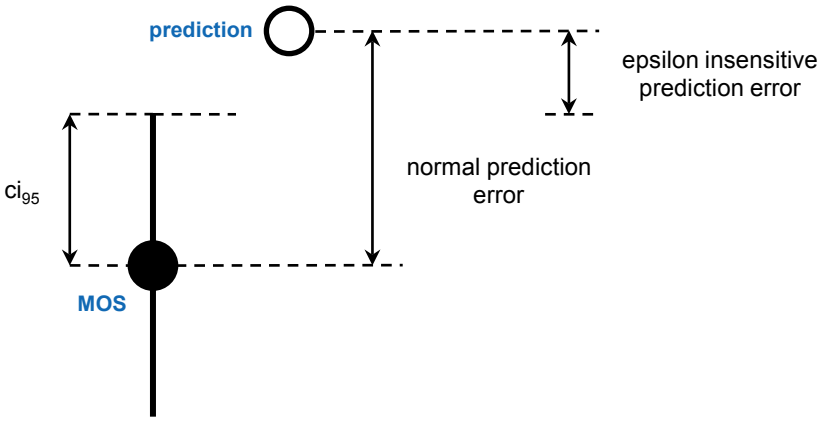

Figure B.1.: Epsilon-insensitive RMSE$_E$: normal prediction error and epsilon-insensitve prediction error (adapted from [110])

and

$$RMSE_I = \frac{RMSE\sqrt{N-1}}{\sqrt{\chi^4_{0.975}(N-1)}}. \tag{B.3.16b}$$

The statistical significance of the difference between the RSME achieved by two models can be determined by using the F-distributed q statistic defined as

$$q = \frac{\mathrm{RMSE}^2_{max}}{\mathrm{RMSE}^2_{min}}, \tag{B.3.17}$$

where RMSE$_{max}$ and RMSE$_{min}$ are the highest and lowest RMSE in the comparison, respectively. For N samples, the value $F(0.05, N, N)$ of the F-distribution for the 95% confidence level can then be compared with q to assess the statistical significance of the difference between the RMSEs.

B.3.5. Epsilon-insensitive RMSE$_E$

The *epsilon-insensitive RMSE* or *RMSE$_E$* is an extension of the RMSE proposed in ITU-T P.1401 [110]. Usually, the visual quality **y** is represented by the MOS of the corresponding sequences. But as already discussed in Section 2.5.2, the MOS itself is only an average of all ratings votes and therefore does not represent the variance of the subjects' ratings. The idea is to take the confidence interval and therefore the uncertainty of the MOS into account when determining the prediction error with the RMSE, as illustrated in Fig. B.1. The epsilon-insensitive prediction error $E_{p,n}$ for the n-th sequence is then determined by

$$E_{p,n} = \max(0, |y_n - \widehat{y}_n| - ci_{95,n}), \tag{B.3.18}$$

where $ci_{95,n}$ is the (one-sided) 95% confidence interval of the MOS and y_n and $\widehat{y}_n$ its visual quality and visual quality estimation, respectively. If the prediction $\widehat{y}_n$ falls within the confidence interval, $E_{p,n}$ will be set to 0, otherwise only the difference between the the confidence interval and the prediction will be included in $E_{p,n}$. The RMSE_E for all N samples is then given by

$$RMSE_E = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(E_{p,n})^2}. \tag{B.3.19}$$

Note, that in order to calculate the RMSE_E either the 95% confidence intervals themselves or the individual subject's ratings must be available. The confidence intervals and the statistical significance of the difference between two RMSEs_E can be determined as for the RSME.

B.3.6. Outlier ratio

The last performance metric used in this thesis is the *outlier ratio* or *OR* as defined in ITU-T P.1410 [110]. This very simple performance metric represents the number of quality predictions $\widehat{y}_n$ that fall outside the 95% confidence interval. With OR_n denoting if the n-th sequence is an outlier as

$$OR_n = \begin{cases} 1 & \text{if } |y_n - \widehat{y}_n| > ci_{95,n} \\ 0 & \text{if } |y_n - \widehat{y}_n| \leq ci_{95,n} \end{cases}, \tag{B.3.20}$$

we can write the outlier ratio for all N sequences as

$$OR = \frac{\sum_{n=1}^{N} OR_n}{N} \tag{B.3.21}$$

where $ci_{95,n}$ is the (one-sided) 95% confidence interval of the MOS and y_n and $\widehat{y}_n$ its visual quality and visual quality estimation, respectively. Similar to the RMSE_E, we also need for the outlier ratio either the 95% confidence intervals or the individual subject's ratings in order to determine the confidence intervals.

Because the OR represents the proportion of outliers in the N samples, the binomial distribution can be used to describe the statistical properties of the OR [110]. The standard deviation σ_{OR} of the OR is then given as

$$\sigma_{OR} = \sqrt{\frac{OR(1-OR)}{N}}. \tag{B.3.22}$$

The upper and lower limit of the 95% confidence interval of the OR, OR_u and OR_l are then provided as

$$OR_u = OR + 1.96\sigma_{OR} \tag{B.3.23a}$$

and

$$OR_l = OR - 1.96\sigma_{OR}, \tag{B.3.23b}$$

where 1.96 is the corresponding standard score for the 95% interval for the normally distributed OR. For $N \leq 30$, Student's t distribution with $N - 1$ degrees of freedom must be used for determining the values corresponding to the (two-tailed) 95% confidence intervals.

In ITU-T P.1401 [110] it is suggested to determine the statistical significance of the difference between two outlier ratios OR_1 and OR_2, representing the outlier ratios of two models for the same data, with the statistic

$$z = \frac{OR_1 - OR_2}{\sigma_{OR_1 - OR_2}}, \tag{B.3.24}$$

where the standard deviation $\sigma_{OR_1 - OR_2}$ of the difference between the two outlier ratios OR_1 and OR_2 for N samples is given by

$$\sigma_{OR_1 - OR_2} = \sqrt{\frac{\sigma_{OR_1}^2}{N} + \frac{\sigma_{OR_2}^2}{N}}. \tag{B.3.25}$$

Using the z-score from (B.3.24), the statistical significance can then be determined as for the Pearson correlation coefficient.

B.4. Data sets

In this section of the appendix, further information about the data sets discussed in Section 9.2 will be provided.

Firstly, the spatial perceptual information (SI) according and temporal perceptual information measurements (TI) according to ITU-T P.910 [121] are given in Fig. B.2. They provided a rough measurement of both the spatial and the temporal complexity of the given video sequences. Secondly, the number of frames and length of the video sequences are

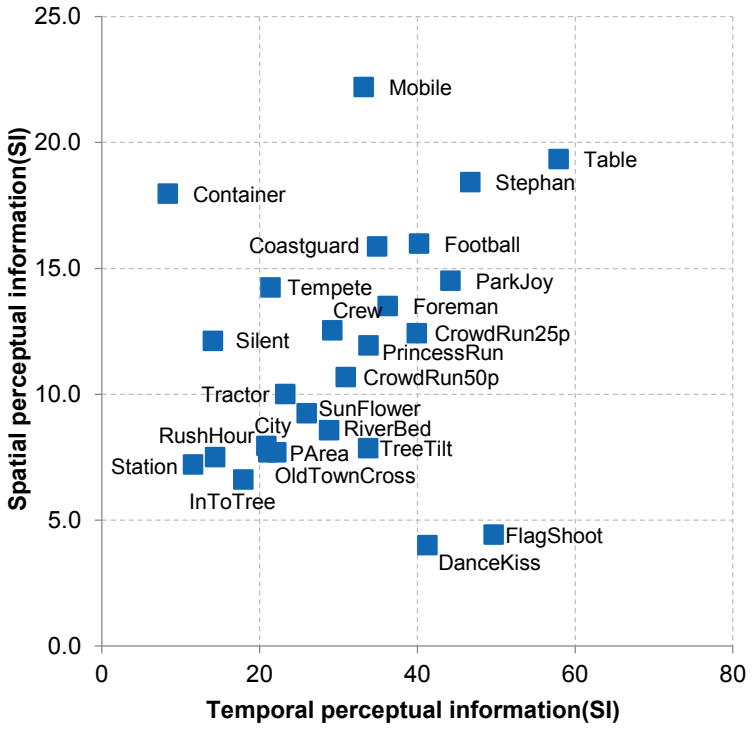

Figure B.2.: Spatial perceptual information (SI) and temporal perceptual information measurements (TI) according to ITU-T P.910 [121] for all video sequences

provided in Table B.2 on the next page. Lastly, the bitrates and corresponding MOS scores for the different rate points (RP) for each of the video squences in Table B.3 on page 247, Table B.4 on page 248 and Table B.5 on page 249.

Table B.2.: Number of frames and length of the video sequences in the data sets used for the performance comparison

Data set	Video sequences	Frames	Length [s]
TUM1080p50	CrowdRun	500	10
	TreeTilt	500	10
	PrincessRun	500	10
	DanceKiss	500	10
	FlagShoot	491	9.82
TUM1080p25	CrowdRun	248[a], 250[b]	9.92[a], 10[b]
	ParkJoy	248[a], 250[b]	9.92[a], 10[b]
	InToTree	248[a], 250[b]	9.92[a], 10[b]
	OldTownCross	248[a], 250[b]	9.92[a], 10[b]
LIVE Video Quality	PedestrianArea	250	10
	RiverBed	250	10
	RushHour	250	10
	Sunflower	250	10
	Station	250	10
	Tractor	250	10
IT-IST Video Quality	Australia	298	11.92[c]
	City	298	9.93
	Coastguard	298	9.93
	Container	298	9.93
	Crew	298	9.93
	Football	259	8.63
	Foreman	298	9.93
	Mobile	298	9.93
	Silent	298	9.93
	Stephan	298	9.93
	Table	298	9.93
	Tempete	259	8.63

[a] High complexity subset
[b] Low complexity subset
[c] Only 25 fps

Table B.3.: Detailed overview of the video sequences from the TUM1080p50/TUM1080p25 data set used in the performance comparison

Data set	Video Sequence	SI[a]	TI[b]	Bitrates [MBit/s]				MOS[c]			
				RP1	RP2	RP3	RP4	RP1	RP2	RP3	RP4
TUM1080p50	CrowdRun	10.68	30.95	8	20	30	40	1.9	5.9	8.0	8.0
	TreeTilt	7.85	33.76	2	3	6	10	3.0	5.6	8.9	8.9
	PrincessRun	11.94	33.82	8	20	30	40	1.9	5.6	7.4	7.0
	DanceKiss	4.00	41.28	2	3	6	10	2.9	5.4	8.2	8.0
	FlagShoot	4.42	49.68	2	3	6	10	2.5	5.5	8.0	7.7
TUM1080p25	CrowdRun	12.42	39.95	8.4	12.7	19.2	28.5	5.0[d]	6.9[d]	8.3[d]	9.3[d]
								2.6[e]	5.6[e]	6.2[e]	7.8[e]
	ParkJoy	14.51	44.22	9.0	12.6	20.1	30.9	6.8[d]	7.5[d]	8.9[d]	9.6[d]
								1.9[e]	2.1[e]	5.4[e]	8.4[e]
	InToTree	6.60	17.94	5.7	10.4	13.1	17.1	7.3[d]	8.6[d]	9.2[d]	9.3[d]
								4.3[e]	6.2[e]	6.3[e]	6.2[e]
	OldTownCross	7.69	21.12	5.4	9.6	13.7	19.0	8.9[d]	9.0[d]	9.4[d]	9.6[d]
								6.9[e]	7.8[e]	7.9[e]	8.2[e]

[a] Spatial perceptual information(SI)according to ITU-T P.910 [121]
[b] Temporal perceptual information measurement (TI) according to ITU-T P.910 [121]
[c] Discrete 11-point scale from 0-10, worst to best
[d] High complexity subset
[e] Low complexity subset

Table B.4.: Detailed overview of the video sequences from the LIVE Video Quality data set used in the performance comparison

Video Sequence	SI[a]	TI[b]	Bitrates [kBit/s]				DMOS[c]			
			RP1	RP2	RP3	RP4	RP1	RP2	RP3	RP4
PedestrianArea	7.94	20.87	252	301	401	601	40.55	52.61	60.25	68.72
RiverBed	8.57	28.81	800	1000	1500	2000	39.19	43.68	55.85	63.58
RushHour	7.49	14.36	202	252	303	403	37.87	45.44	53.63	62.99
Sunflower	9.24	25.97	305	357	404	610	32.60	44.01	54.94	57.15
Station	7.21	11.58	452	503	555	606	40.77	46.56	52.33	56.08
Tractor	10.0	23.27	452	500	601	800	38.67	47.77	56.91	63.79

[a] Spatial perceptual information(SI)according to ITU-T P.910 [121]
[b] Temporal perceptual information measurement (TI) according to ITU-T P.910 [121]
[c] Continuous DMOS scale from 0-100, worst to best

Table B.5.: Detailed overview of the video sequences from the IT-IST Video Quality data set used in the performance comparison

Video Sequence	SI[a]	TI[b]	Bitrates [kBit/s]						MOS[c]					
			RP1	RP2	RP3	RP4	RP5	RP6	RP1	RP2	RP3	RP4	RP5	RP6
Australia[d]	17.60	9.52	32	64	128	256	–	–	2.22	3.50	4.50	4.94	–	–
City	10.69	22.11	128	200	256	512	–	–	3.26	3.63	4.21	4.89	–	–
Coastguard	15.86	34.91	65	100[e]	131	200[e]	262	525	1.83	2.79[e]	3.26	3.89[e]	4.39	4.68
Container	17.96	8.70	65	131	262	524	–	–	3.67	3.72	4.78	4.94	–	–
Crew	12.54	29.23	127	199	400	1024	–	–	1.26	2.05	3.74	4.95	–	–
Football	15.98	40.24	263	401[e]	526	756[e]	1050	2105	1.72	2.58[e]	3.22	3.79[e]	3.95	4.94
Foreman	13.50	35.25	66	131	263	525	–	–	1.06	2.83	4.17	4.89	–	–
Mobile	22.19	33.21	126	131	202[e]	262	400[e]	524	1.28	1.44	2.58[e]	3.94	3.95[e]	4.68
Silent	12.11	14.08	64	200	400	1025	–	–	1.58	3.79	4.58	5.00	–	–
Stephan	18.42	46.74	140	200[e]	263	401[e]	525	1050	1.00	2.11[e]	2.58	3.63[e]	4.28	4.95
Table	19.33	57.91	66	132	263	525	–	–	1.22	2.83	4.50	4.89	–	–
Tempete	14.24	21.39	130	202	405	756	–	–	2.58	3.53	4.42	4.95	–	–

[a] Spatial perceptual information(SI)according to ITU-T P.910 [121]
[b] Temporal perceptual information measurement (TI) according to ITU-T P.910 [121]
[c] Discrete 5-point scale from 1–5, worst to best
[d] Only used for bitstream-based example metric
[e] Rate point not used in this thesis

B.5. Additional results for the bitstream-based example metric

This section provides additional results to the performance comparison of the two-way and multi-way data analysis methods for the bitstream-based example metric in Section 9.3.1. In particular the PRESS and scatter plots for the TUM1080p25, TUM1080p50 and LIVE data sets are presented in this section.

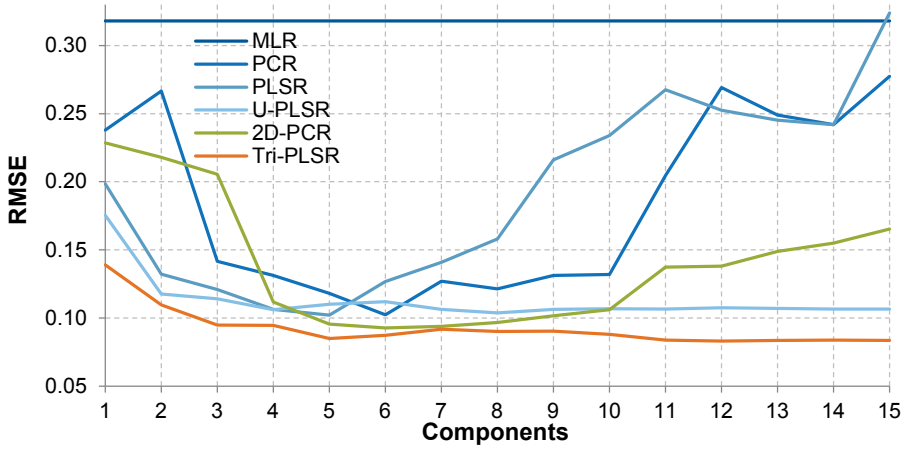

Figure B.3.: PRESS plot for the bitstream-based example metric and the TUM1080p25 data set in order to determine the optimal number of R components for the prediction error as expressed by the RMSE for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

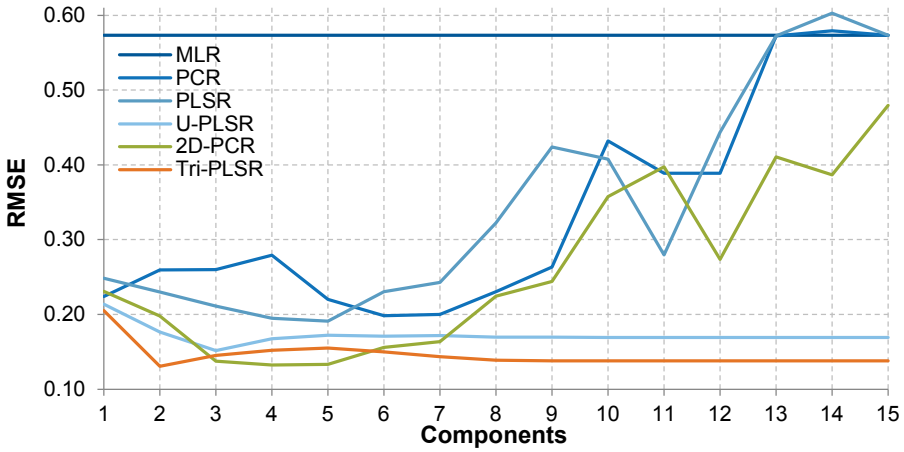

Figure B.4.: PRESS plot for the bitstream-based example metric and the TUM1080p50 data set in order to determine the optimal number of *R* components for the prediction error as expressed by the RMSE for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

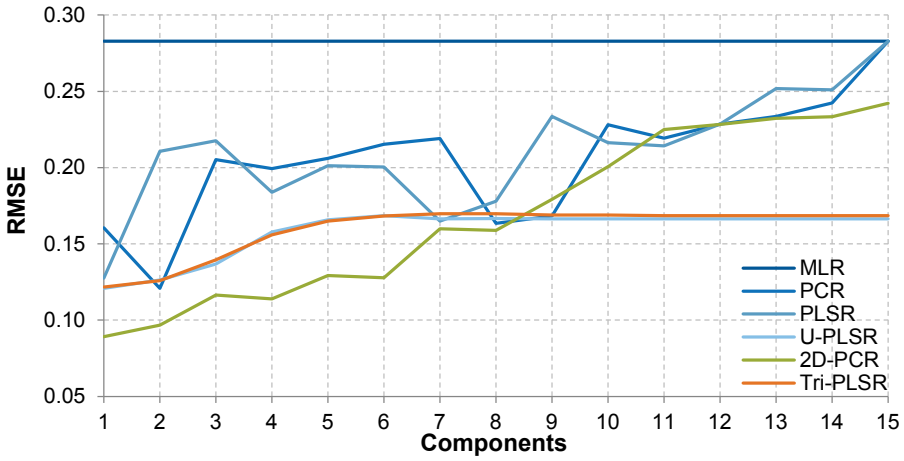

Figure B.5.: PRESS plot for the bitstream-based example metric and the LIVE data set in order to determine the optimal number of *R* components for the prediction error as expressed by the RMSE for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

Table B.6.: Prediction performance of the bitstream-based example metric for all data sets and MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR, and trilinear PLSR (Tri-PLSR) for R components with respect to the Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive RMSE_E, where the OR, RMSE and RMSE_E are expressed as their complement to the maximum value of 1

Metric	MLR	PCR	PLSR	U-PLSR	2D-PCR	Tri-PLSR
IT-IST, R = 2						
r_p	0.779	0.910	0.926	0.942	0.923	0.951
r_s	0.761	0.895	0.904	0.941	0.931	0.962
r_k	0.585	0.740	0.741	0.784	0.777	0.845
OR	0.375	0.396	0.458	0.396	0.271	0.417
RMSE	0.792	0.863	0.876	0.884	0.855	0.892
RMSE_E	0.835	0.910	0.924	0.932	0.903	0.940
TUM1080p25, R = 5						
r_p	0.395	0.835	0.877	0.858	0.898	0.920
r_s	0.351	0.788	0.838	0.856	0.874	0.888
r_k	0.281	0.591	0.660	0.688	0.712	0.761
OR	0.594	0.844	0.938	0.906	0.969	1.000
RMSE	0.682	0.882	0.898	0.890	0.904	0.915
RMSE_E	0.765	0.974	0.995	0.995	0.998	1.000
TUM1080p50, R = 2						
r_p	0.013	0.317	0.536	0.752	0.531	0.839
r_s	0.014	0.269	0.487	0.623	0.457	0.701
r_k	0.011	0.221	0.389	0.421	0.274	0.505
OR	0.150	0.450	0.700	0.700	0.550	0.700
RMSE	0.427	0.740	0.770	0.824	0.802	0.869
RMSE_E	0.557	0.837	0.854	0.910	0.900	0.948
LIVE, R = 1						
r_p	0.327	0.009	0.380	0.377	0.361	0.383
r_s	0.370	0.297	0.459	0.554	0.530	0.537
r_k	0.304	0.174	0.333	0.406	0.435	0.377
OR	0.000	0.000	0.000	0.000	0.000	0.000
RMSE	0.717	0.840	0.872	0.879	0.911	0.878
RMSE_E	0.717	0.840	0.872	0.879	0.911	0.878

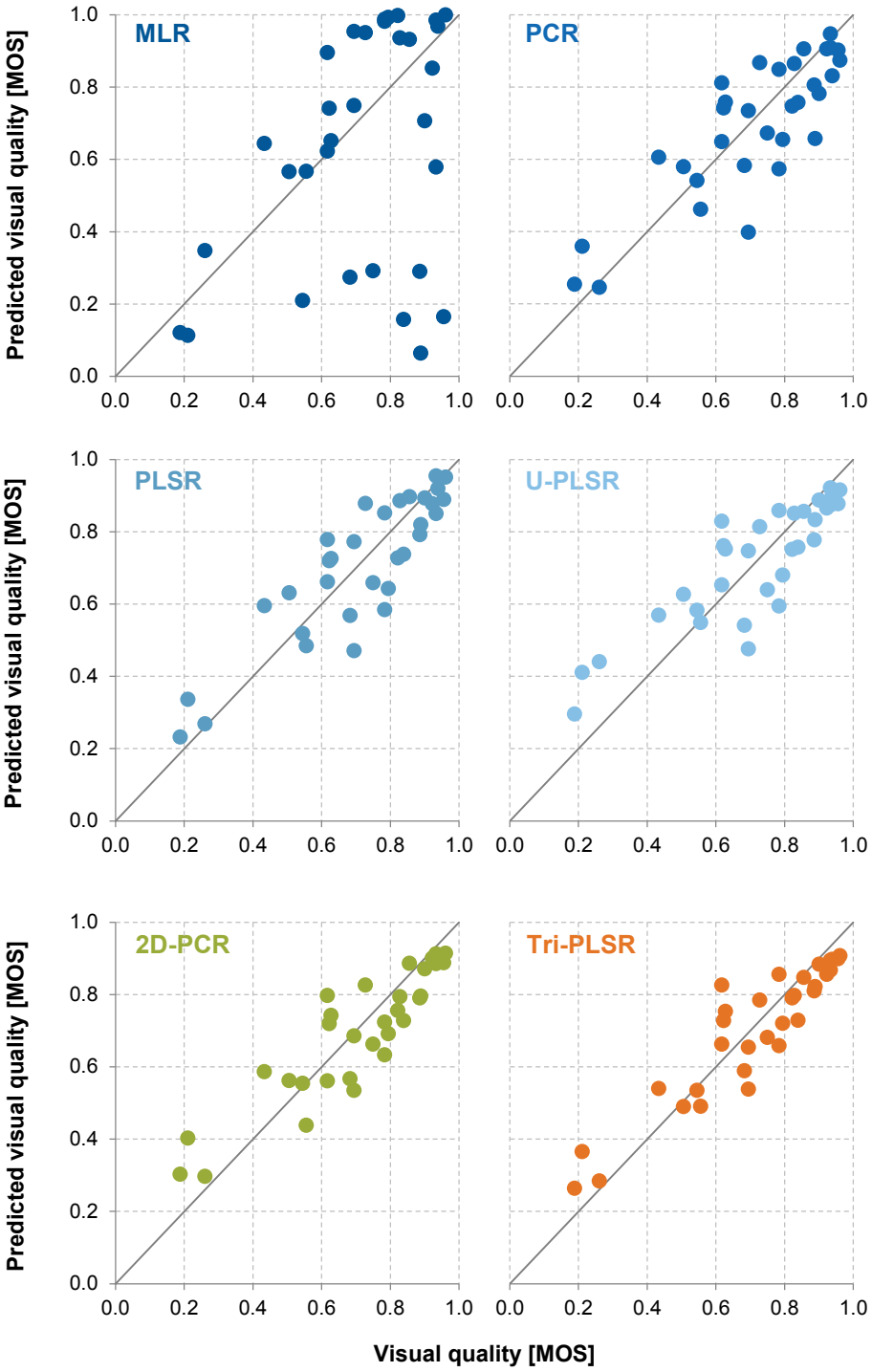

Figure B.6.: Scatter plots of the prediction results of the bitstream-based example metric for the TUM1080p25 data set, showing the visual quality y the against predicted visual quality $\widehat{y}$ for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

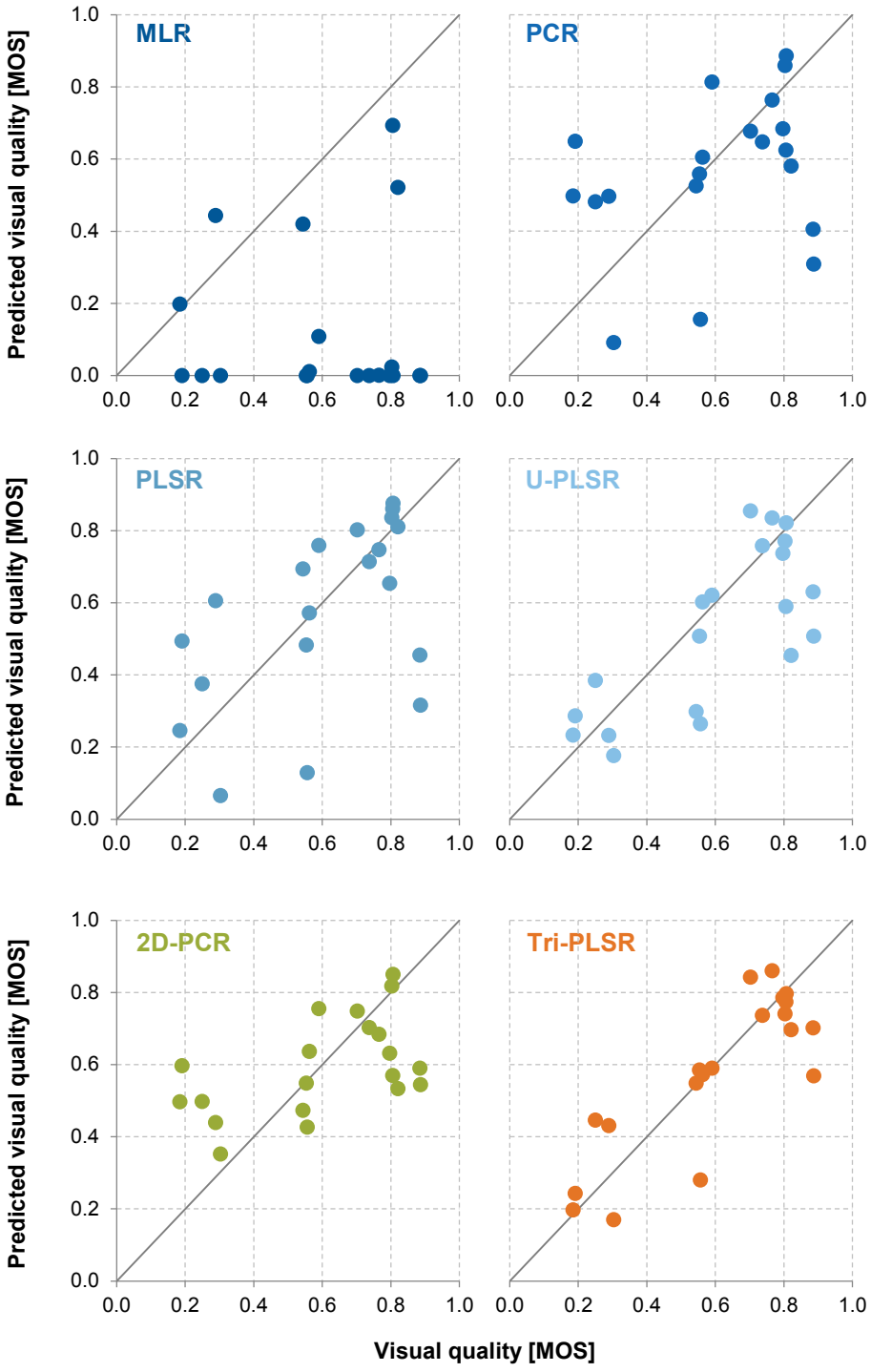

Figure B.7.: Scatter plots of the prediction results of the bitstream-based example metric for the TUM1080p50 data set, showing the visual quality y the against predicted visual quality $\widehat{y}$ for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

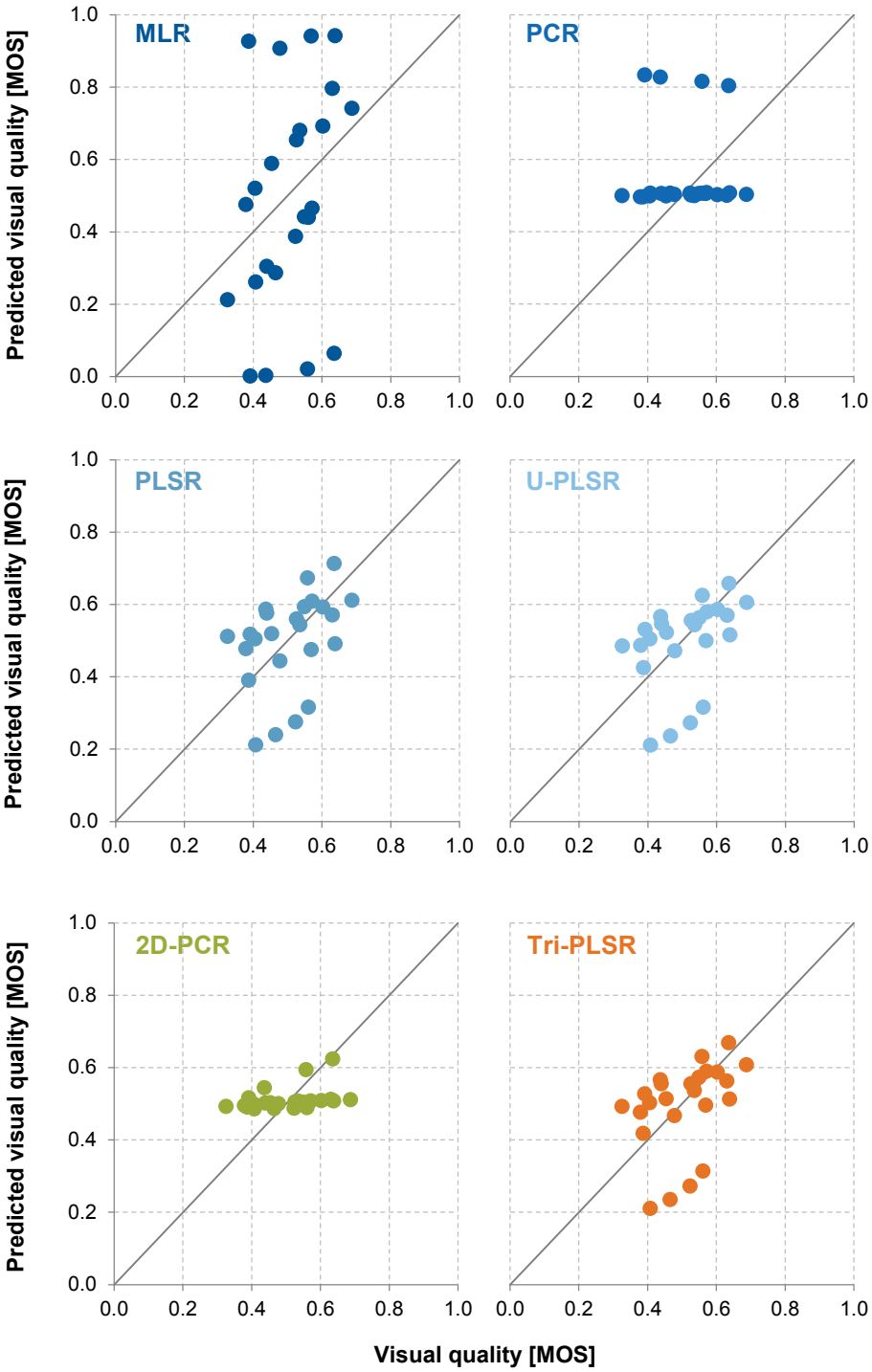

Figure B.8.: Scatter plots of the prediction results of the bitstream-based example metric for the LIVE data set, showing the visual quality y the against predicted visual quality $\widehat{y}$ for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

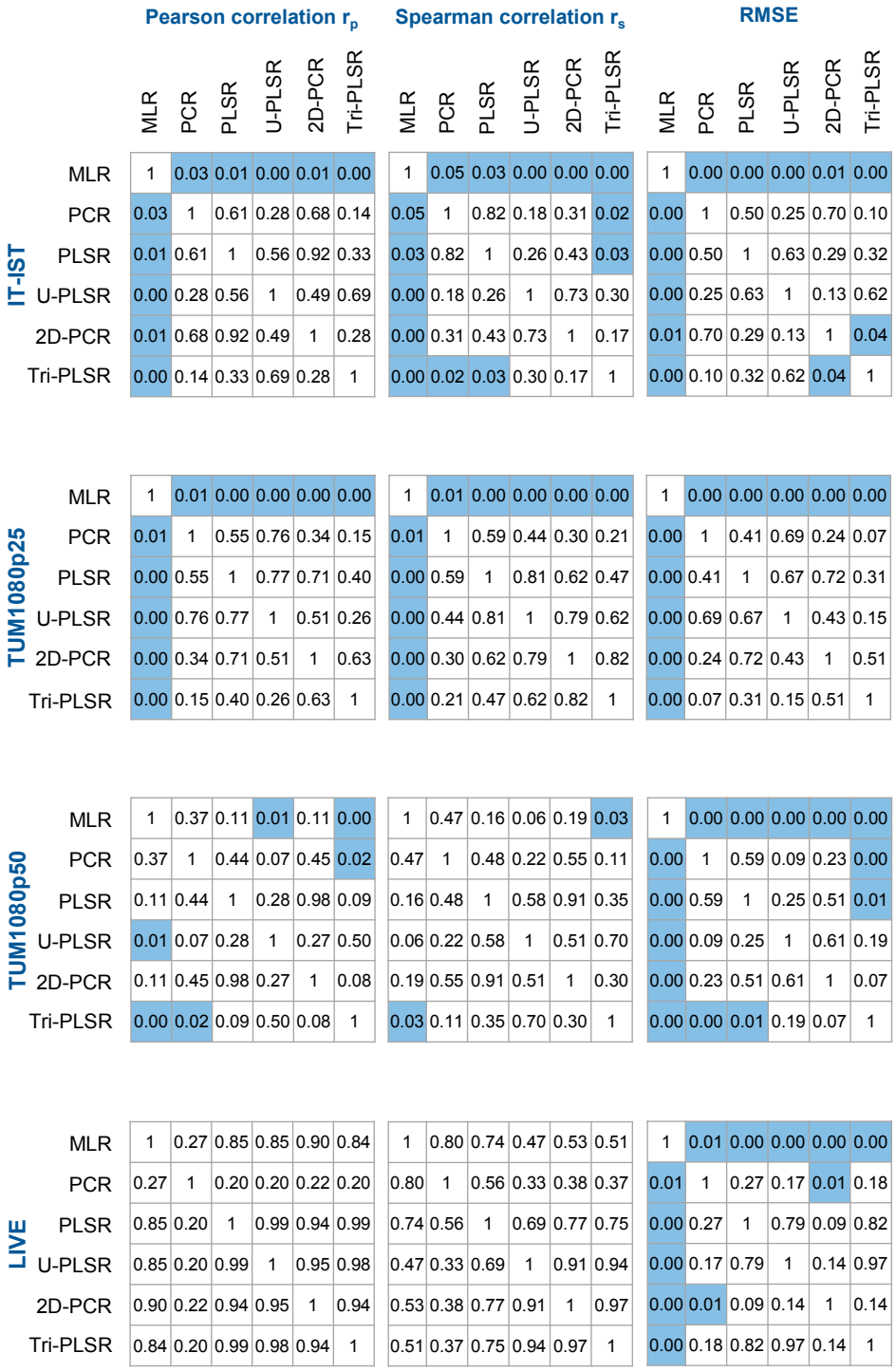

Figure B.9.: Statistical significance of the difference between prediction results for the bitstream-based metric built with different data analysis methods. For each combination the p-value is provided and results that are statistical significant at the 0.05 level with $p < 0.05$ are highlighted

B.6. Additional results for the pixel-based example metric

This section provides additional results to the performance comparison of the two-way and multi-way data analysis methods for the pixel-based example metric in Section 9.3.2. In particular the PRESS and scatter plots for the TUM1080p25, TUM1080p50 and LIVE data sets are presented in this section.

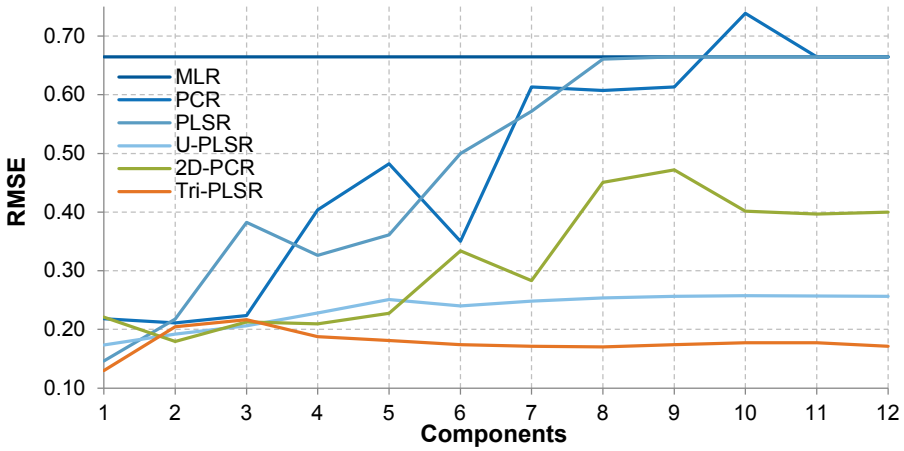

Figure B.10.: PRESS plot for the pixel-based example metric and the TUM1080p25 data set in order to determine the optimal number of R components for the prediction error as expressed by the RMSE for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR). For clarity only the first 15 components are shown

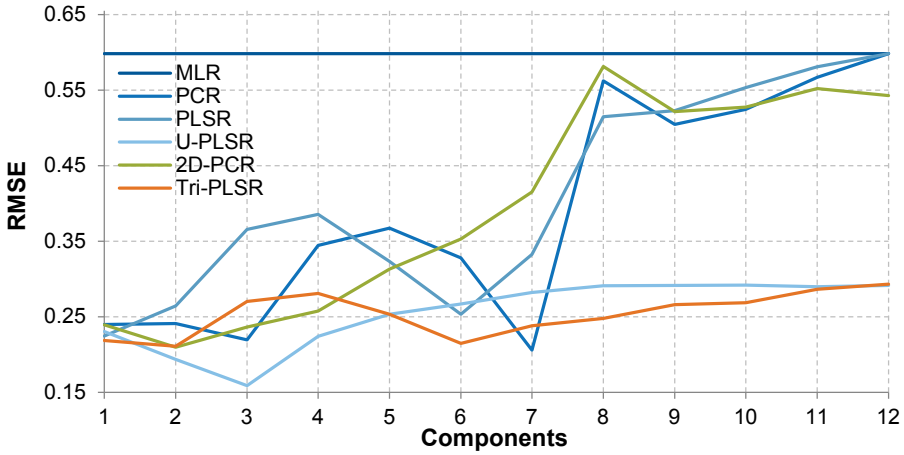

Figure B.11.: PRESS plot for the pixel-based example metric and the TUM1080p50 data set in order to determine the optimal number of R components for the prediction error as expressed by the RMSE for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR). For clarity only the first 15 components are shown

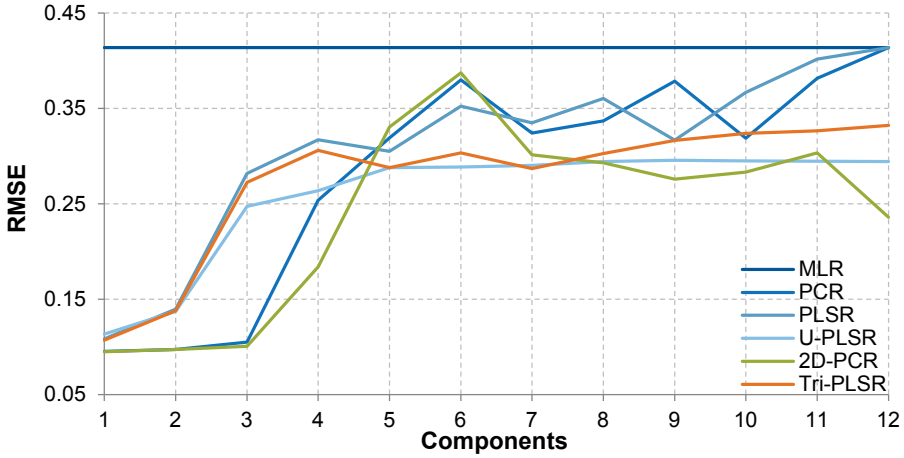

Figure B.12.: PRESS plot for the pixel-based example metric and the LIVE data set in order to determine the optimal number of R components for the prediction error as expressed by the RMSE for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR). For clarity only the first 15 components are shown

Table B.7.: Prediction performance of the pixel-based example metric for all data sets and MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR, and trilinear PLSR (Tri-PLSR) for R components with respect to the Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive RMSE_E, where the OR, RMSE and RMSE_E are expressed as their complement to the maximum value of 1

Metric	MLR	PCR	PLSR	U-PLSR	2D-PCR	Tri-PLSR
IT-IST, R = 6						
r_p	0.752	0.813	0.784	0.629	0.784	0.876
r_s	0.727	0.763	0.740	0.631	0.776	0.834
r_k	0.560	0.607	0.573	0.458	0.602	0.657
OR	0.205	0.295	0.318	0.205	0.205	0.295
RMSE	0.782	0.809	0.793	0.734	0.795	0.841
RMSE_E	0.828	0.856	0.836	0.787	0.843	0.891
TUM1080p25, R = 1						
r_p	0.017	0.044	0.750	0.601	0.175	0.795
r_s	0.146	0.077	0.757	0.518	0.214	0.789
r_k	0.102	0.079	0.559	0.365	0.143	0.591
OR	0.125	0.563	0.844	0.688	0.531	0.781
RMSE	0.335	0.782	0.854	0.827	0.779	0.870
RMSE_E	0.487	0.891	0.958	0.939	0.888	0.965
TUM1080p50, R = 3						
r_p	0.005	0.593	0.277	0.755	0.488	0.614
r_s	0.035	0.668	0.307	0.770	0.608	0.606
r_k	0.011	0.505	0.211	0.611	0.442	0.453
OR	0.000	0.300	0.250	0.700	0.250	0.450
RMSE	0.407	0.786	0.639	0.846	0.769	0.735
RMSE_E	0.546	0.908	0.764	0.931	0.898	0.844
LIVE, R = 1						
r_p	0.303	0.096	0.080	0.050	0.136	0.088
r_s	0.233	0.131	0.195	0.107	0.200	0.207
r_k	0.167	0.123	0.152	0.101	0.174	0.181
OR	0.000	0.000	0.000	0.000	0.000	0.000
RMSE	0.586	0.905	0.892	0.887	0.905	0.893
RMSE_E	0.586	0.905	0.892	0.887	0.905	0.893

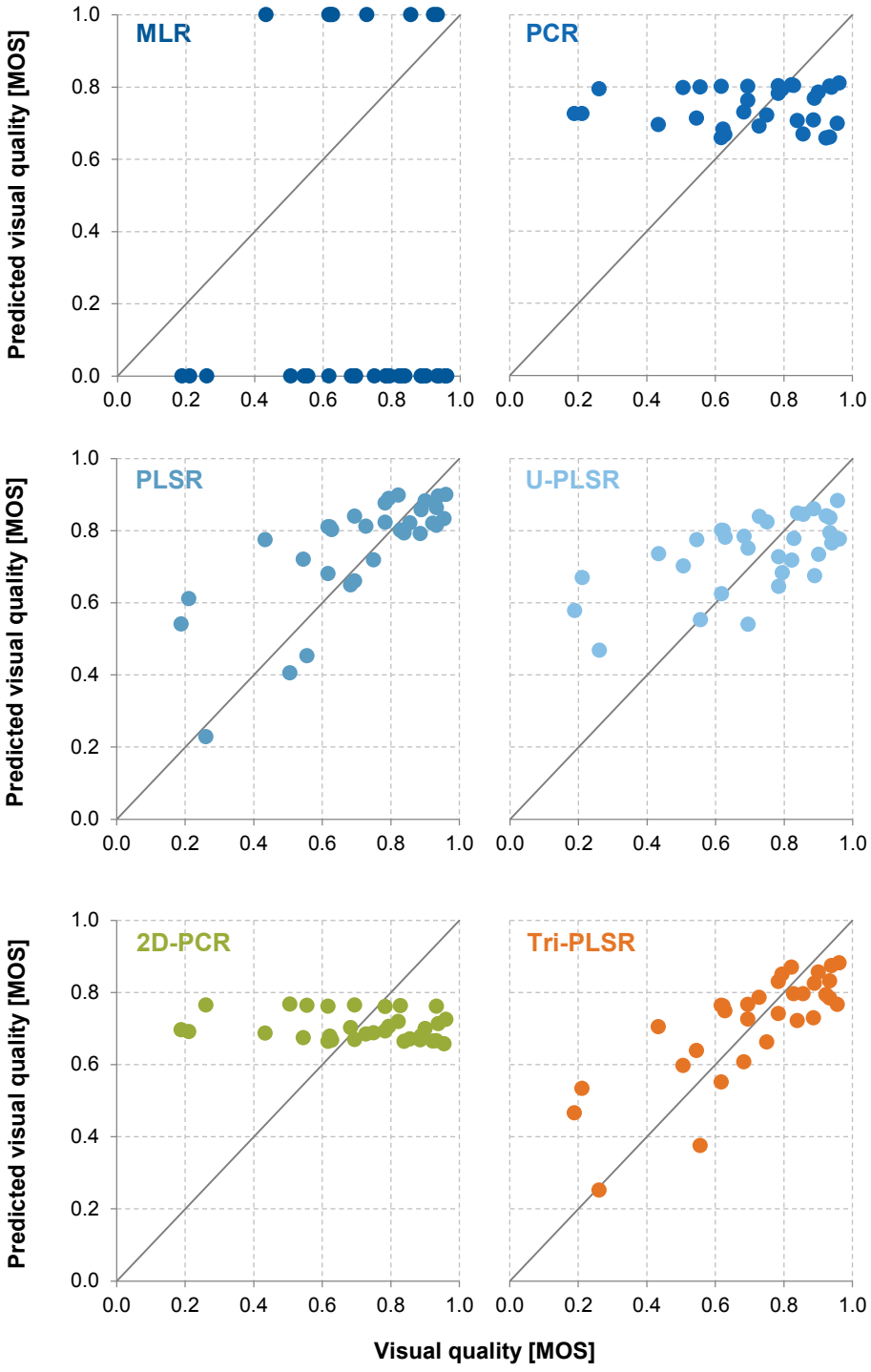

Figure B.13.: Scatter plots of the prediction results of the pixel-based example metric for the TUM1080p25 data set, showing the visual quality y the against predicted visual quality $\widehat{y}$ for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

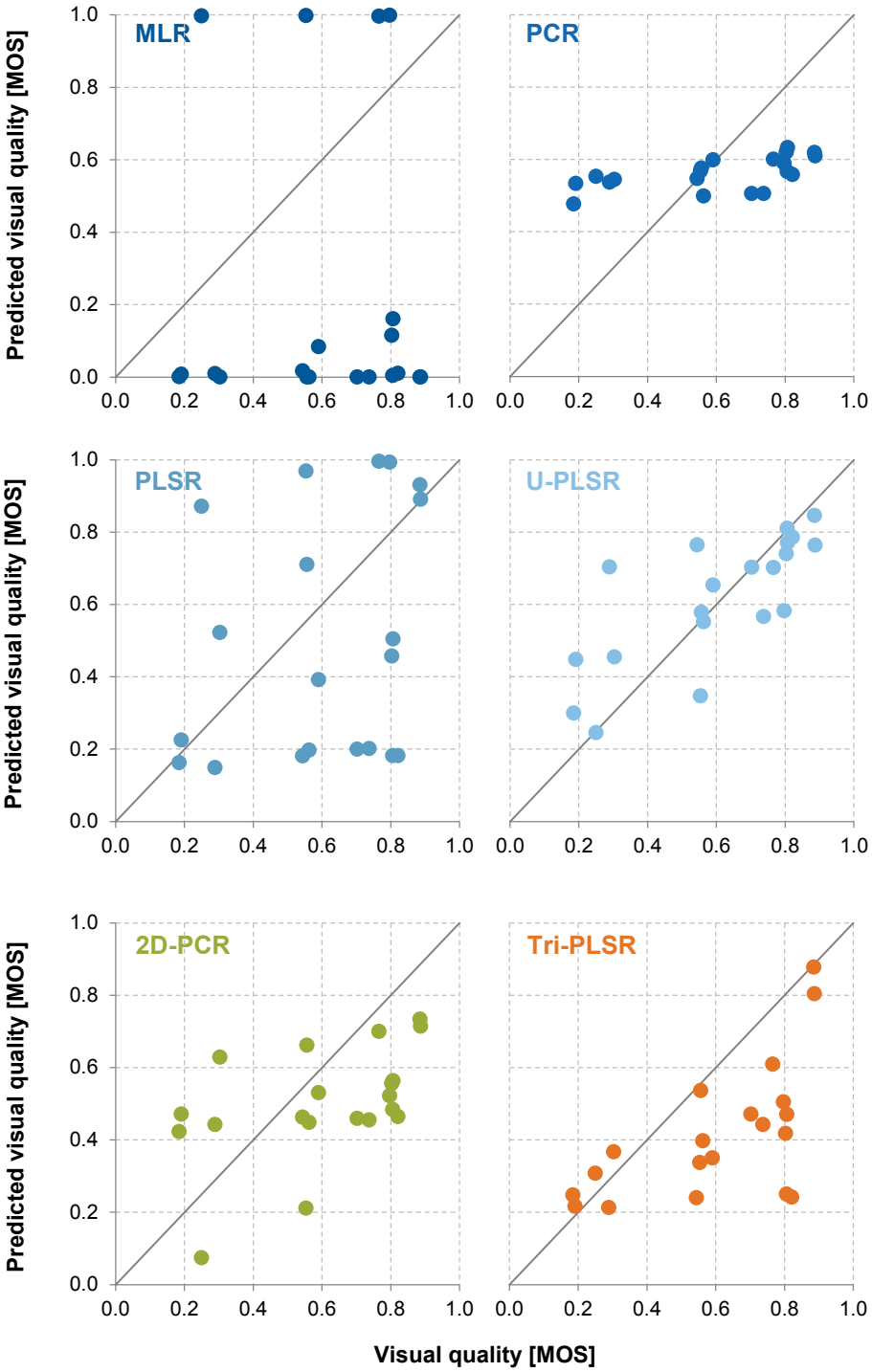

Figure B.14.: Scatter plots of the prediction results of the pixel-based example metric for the TUM1080p50 data set, showing the visual quality y the against predicted visual quality $\widehat{y}$ for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

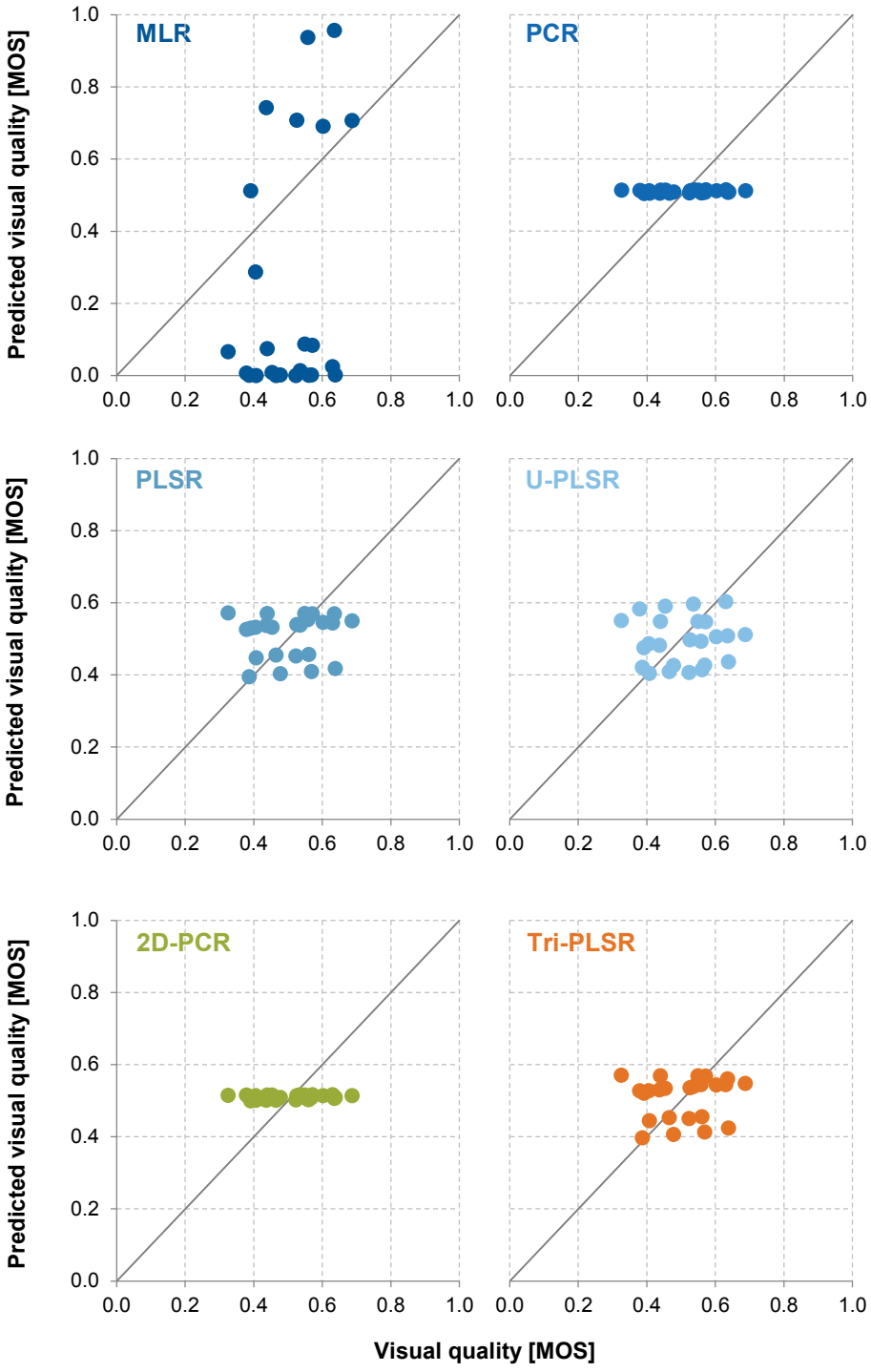

Figure B.15.: Scatter plots of the prediction results of the pixel-based example metric for the LIVE data set, showing the visual quality y the against predicted visual quality $\widehat{y}$ for MLR, PCR, PLSR, unfolding with PLSR (U-PLSR), 2D-PCR and trilinear PLSR (Tri-PLSR)

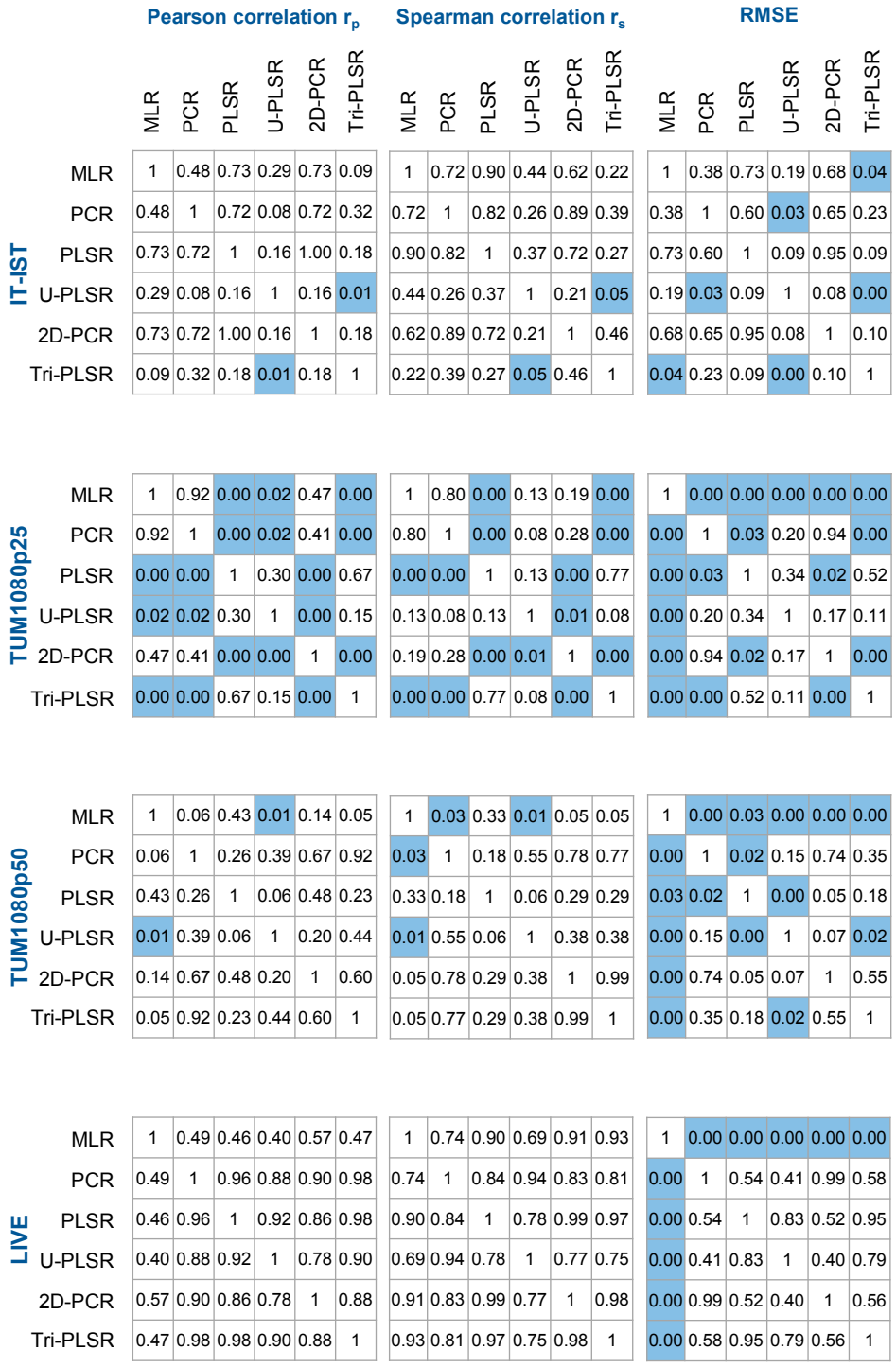

Figure B.16.: Statistical significance of the difference between prediction results for the pixel-based metric built with different data analysis methods. For each combination the *p*-value is provided and results that are statistical significant at the 0.05 level with $p < 0.05$ are highlighted

B.7. Additional results for the comparison to the state-of-the-art – linear data fitting

This section provides additional results to the performance comparison of the bitstream-based example metric built with trilinear PLSR with the selected state-of-the-art metrics in Section 9.4. All state-of-the-art metrics were fit to the data with a linear function. In particular scatter plots for the TUM1080p25, TUM1080p50 and LIVE data sets are presented in this section.

Table B.8.: Prediction performance of the bitstream-based example metric built with trilinear PLSR (Tri-PLSR) and R components compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE metrics and linear data fitting with respect to the Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive RMSE_E, where the OR, RMSE and RMSE_E are expressed as their complement to the maximum value of 1

Metric	PSNR	SSIM	IWSSIM	NIQE	MOVIE	Tri-PLSR
IT-IST, R = 2						
r_p	0.791	0.849	0.914	0.556	0.882	0.951
r_s	0.823	0.871	0.916	0.533	0.902	0.962
r_k	0.647	0.708	0.757	0.389	0.734	0.845
OR	0.396	0.375	0.417	0.188	0.300	0.417
RMSE	0.804	0.830	0.870	0.733	0.844	0.892
RMSE_E	0.847	0.874	0.918	0.779	0.887	0.940
TUM1080p25, R = 5						
r_p	0.718	0.855	0.882	0.147	-	0.920
r_s	0.692	0.816	0.842	0.101	-	0.888
r_k	0.535	0.632	0.660	0.083	-	0.761
OR	0.781	0.844	0.844	0.594	-	1.000
RMSE	0.853	0.891	0.900	0.791	-	0.915
RMSE_E	0.954	0.988	0.990	0.898	-	1.000
TUM1080p50, R = 2						
r_p	0.504	0.680	0.885	0.096	-	0.839
r_s	0.423	0.480	0.779	0.223	-	0.701
r_k	0.347	0.316	0.621	0.179	-	0.505
OR	0.500	0.750	0.900	0.400	-	0.700
RMSE	0.798	0.829	0.891	0.768	-	0.869
RMSE_E	0.899	0.913	0.972	0.879	-	0.948
LIVE, R = 1						
r_p	0.476	0.434	0.627	0.168	0.742	0.383
r_s	0.509	0.525	0.649	0.210	0.671	0.537
r_k	0.384	0.406	0.500	0.152	0.514	0.377
OR	0.000	0.000	0.000	0.000	0.000	0.000
RMSE	0.916	0.914	0.926	0.906	0.936	0.878
RMSE_E	0.916	0.914	0.926	0.906	0.936	0.878

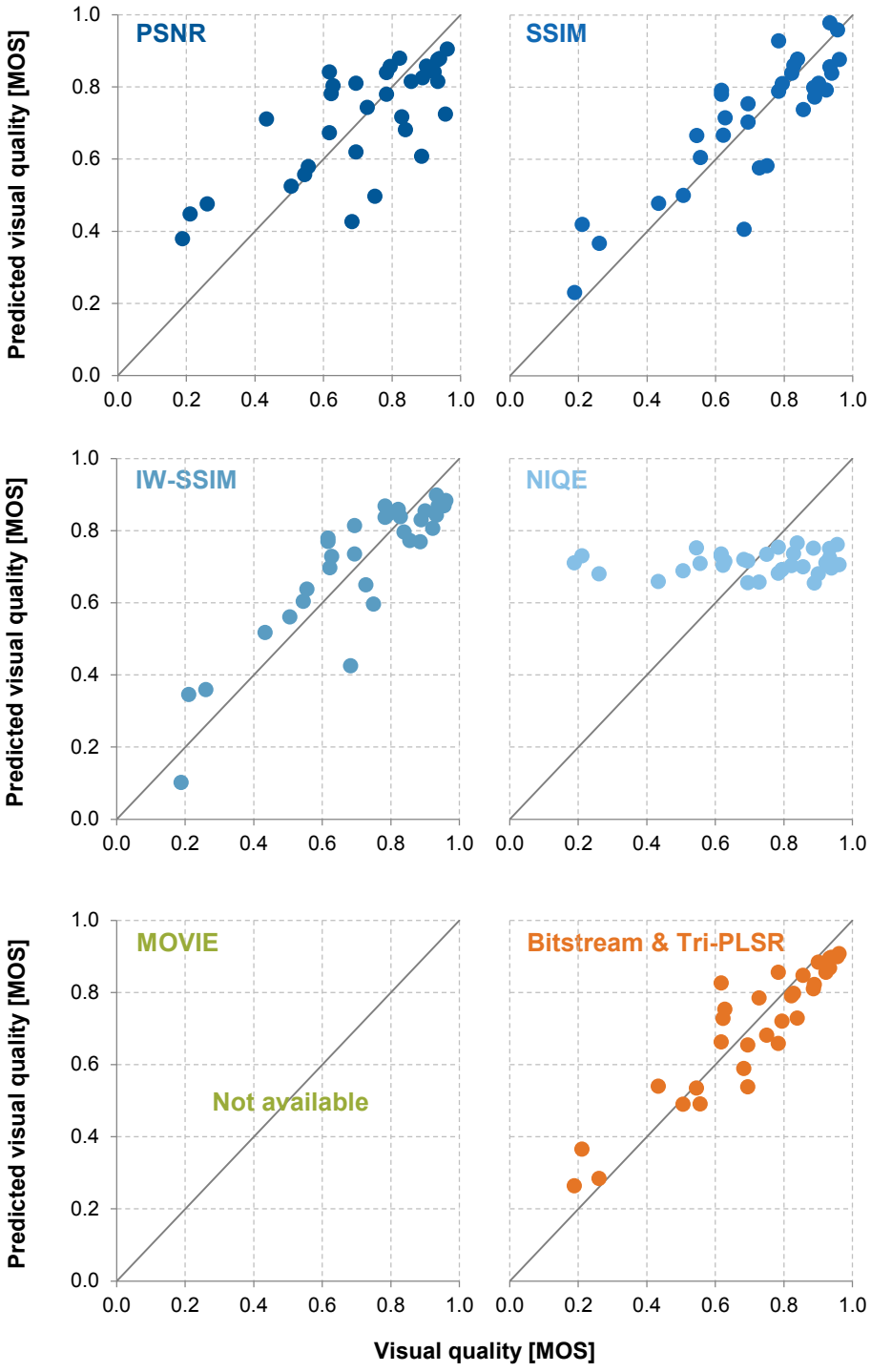

Figure B.17.: Scatter plots of the prediction results of the bitstream-based example metric built with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE for the TUM1080p25 data set, showing the visual quality y against the predicted visual quality $\widehat{y}$

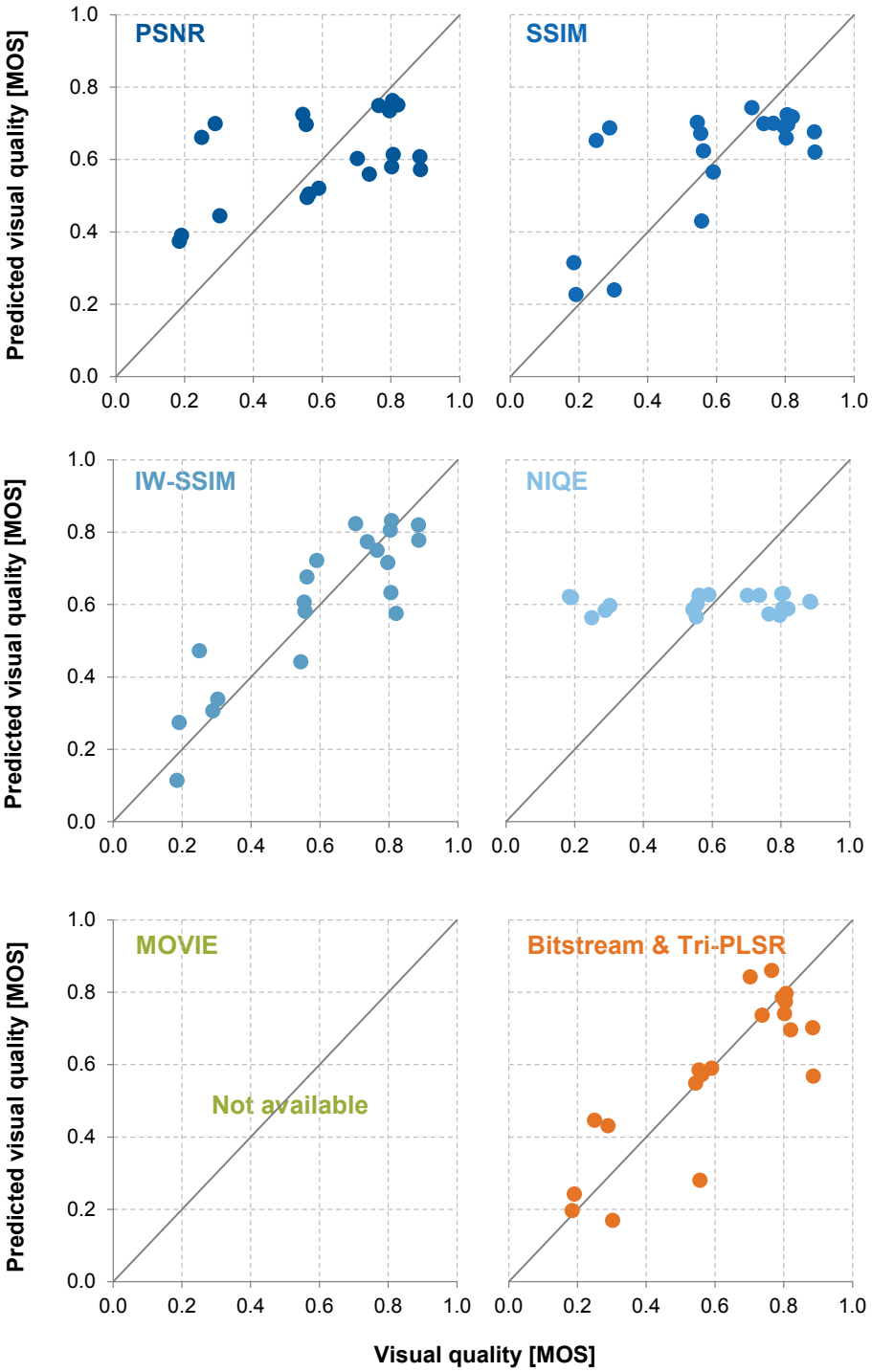

Figure B.18.: Scatter plots of the prediction results of the bitstream-based example metric built with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE for the TUM1080p50 data set, showing the visual quality y against the predicted visual quality $\widehat{y}$

Figure B.19.: Scatter plots of the prediction results of the bitstream-based example metric built with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE for the LIVE data set, showing the visual quality y against the predicted visual quality $\widehat{y}$

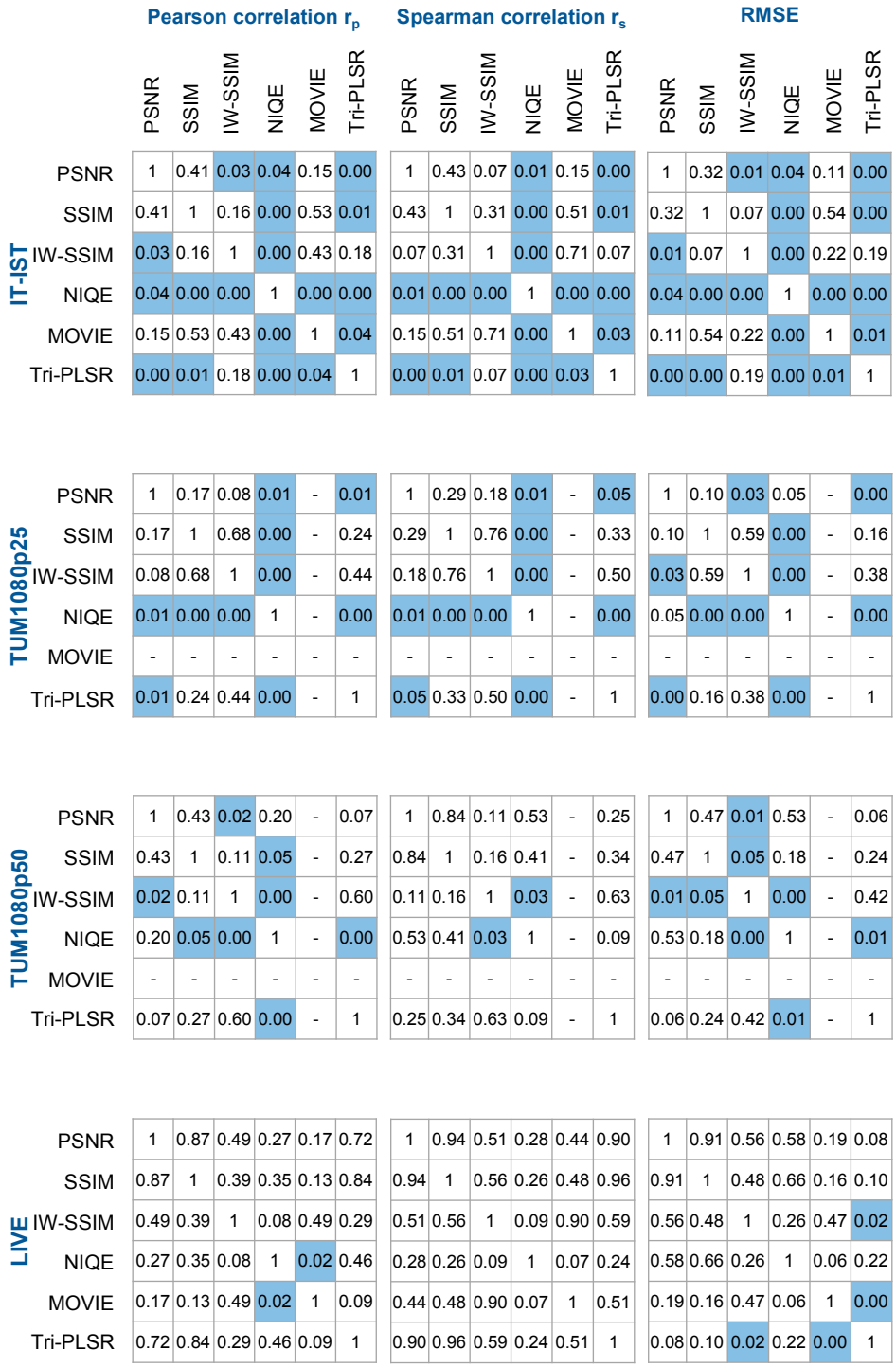

Figure B.20.: Statistical significance of the difference between prediction results of selected metrics and the bitstream-based metric with trilinear PLSR (Tri-PLSR). For each combination the *p*-value is provided and results that are statistical significant at the 0.05 level with $p < 0.05$ are highlighted

B.8. Results for the comparison to the state-of-the-art – cubic data fitting

This section provides the results of a performance comparison of the bitstream-based example metric built with trilinear PLSR with selected state-of-the-art metrics after cubic data fitting, in contrast to Section 9.4, where all state-of-the-art metrics were fit to the data with a linear function.

Table B.9.: Prediction performance of the bitstream-based example metric built with trilinear PLSR (Tri-PLSR) and R components compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE metrics and cubic data fitting with respect to the Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive RMSE_E, where the OR, RMSE and RMSE_E are expressed as their complement to the maximum value of 1

Metric	PSNR	SSIM	IWSSIM	NIQE	MOVIE	Tri-PLSR
IT-IST, R = 2						
r_p	0.807	0.860	0.943	0.560	0.923	0.951
r_s	0.821	0.871	0.916	0.533	0.902	0.962
r_k	0.640	0.704	0.759	0.389	0.736	0.845
OR	0.375	0.375	0.542	0.208	0.500	0.417
RMSE	0.810	0.836	0.893	0.734	0.873	0.892
RMSE_E	0.855	0.880	0.936	0.779	0.917	0.940
TUM1080p25, R = 5						
r_p	0.744	0.858	0.887	0.255	-	0.920
r_s	0.692	0.816	0.842	0.199	-	0.888
r_k	0.535	0.632	0.660	0.139	-	0.761
OR	0.844	0.875	0.875	0.594	-	1.000
RMSE	0.859	0.892	0.902	0.796	-	0.915
RMSE_E	0.958	0.987	0.989	0.904	-	1.000
TUM1080p50, R = 2						
r_p	0.714	0.682	0.893	0.345	-	0.839
r_s	0.698	0.480	0.779	0.251	-	0.701
r_k	0.558	0.316	0.621	0.221	-	0.505
OR	0.750	0.800	0.900	0.600	-	0.700
RMSE	0.837	0.829	0.895	0.781	-	0.869
RMSE_E	0.927	0.913	0.976	0.877	-	0.948
LIVE, R = 1						
r_p	0.590	0.626	0.673	0.176	0.752	0.383
r_s	0.663	0.529	0.643	0.064	0.656	0.537
r_k	0.464	0.391	0.493	0.051	0.493	0.377
OR	0.000	0.000	0.000	0.000	0.000	0.000
RMSE	0.923	0.925	0.929	0.906	0.937	0.878
RMSE_E	0.923	0.925	0.929	0.906	0.937	0.878

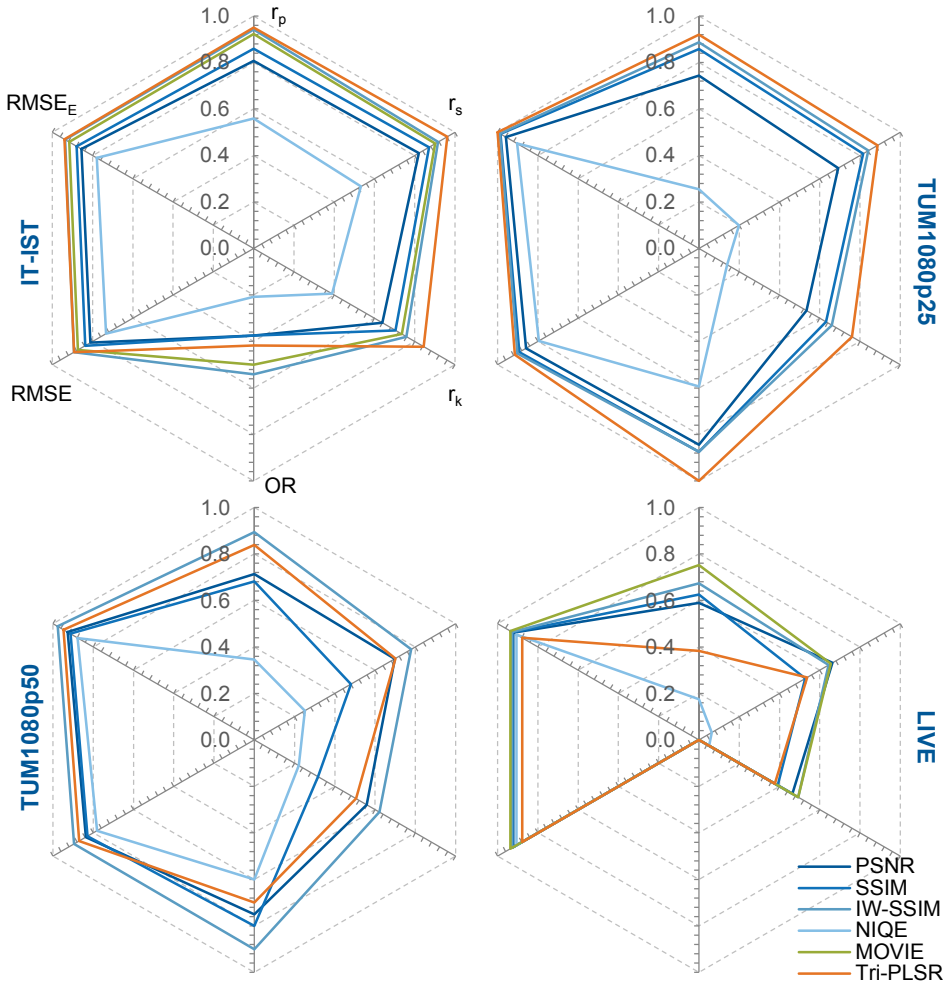

Figure B.21.: Prediction performance for the bitstream-based example metric built with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE metrics and cubic data fitting with respect to the Pearson correlation r_p, Spearman rank correlation r_s, Kendall rank correlation r_k, outlier ration OR, RMSE and epsilon-insensitive $RMSE_E$. For clarity only the axes for the IT-IST data set are labelled

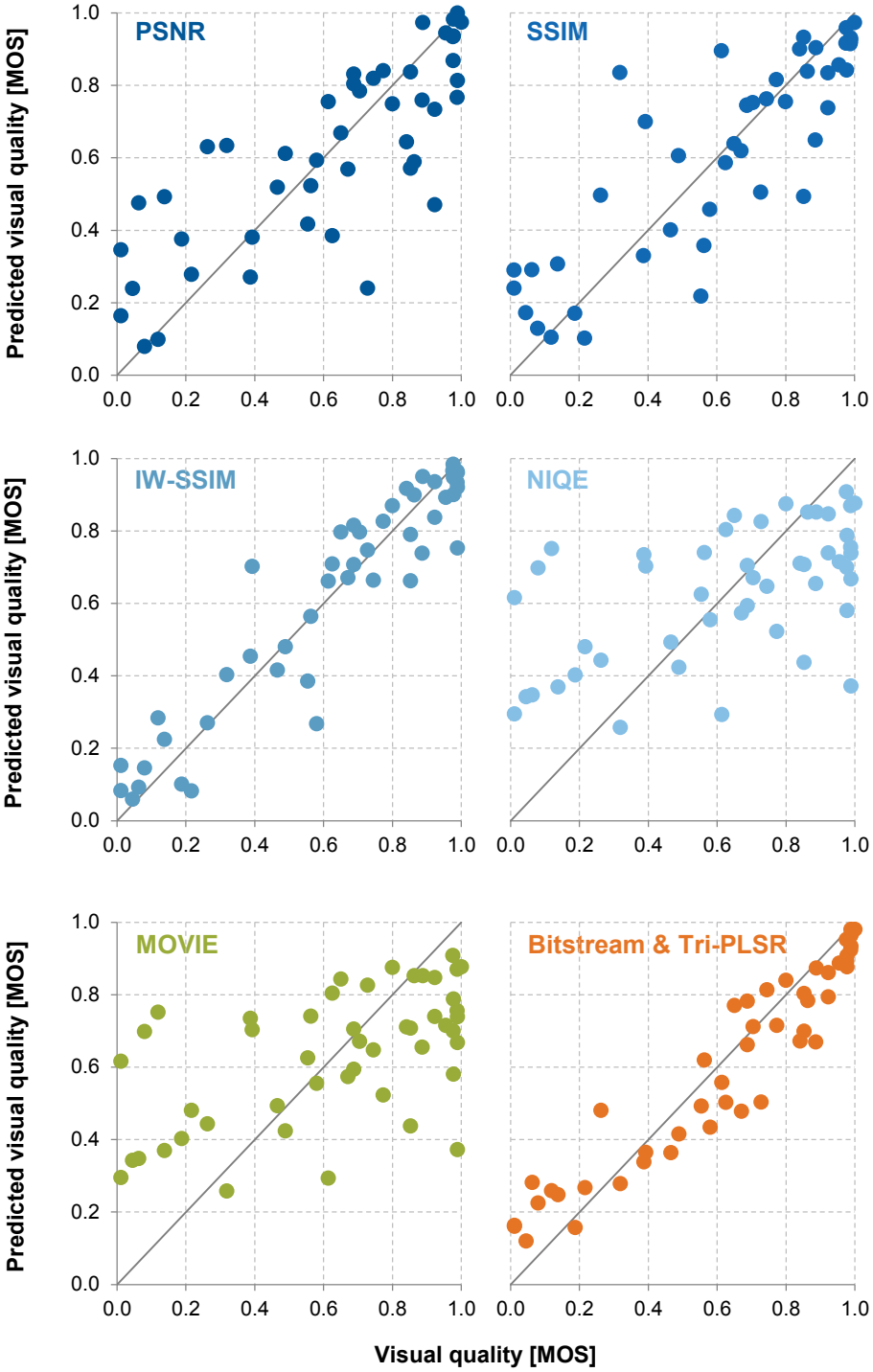

Figure B.22.: Scatter plots of the prediction results of the bitstream-based example metric with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE with cubic data fitting for the IT-IST data set, showing the visual quality y against the predicted quality $\widehat{y}$

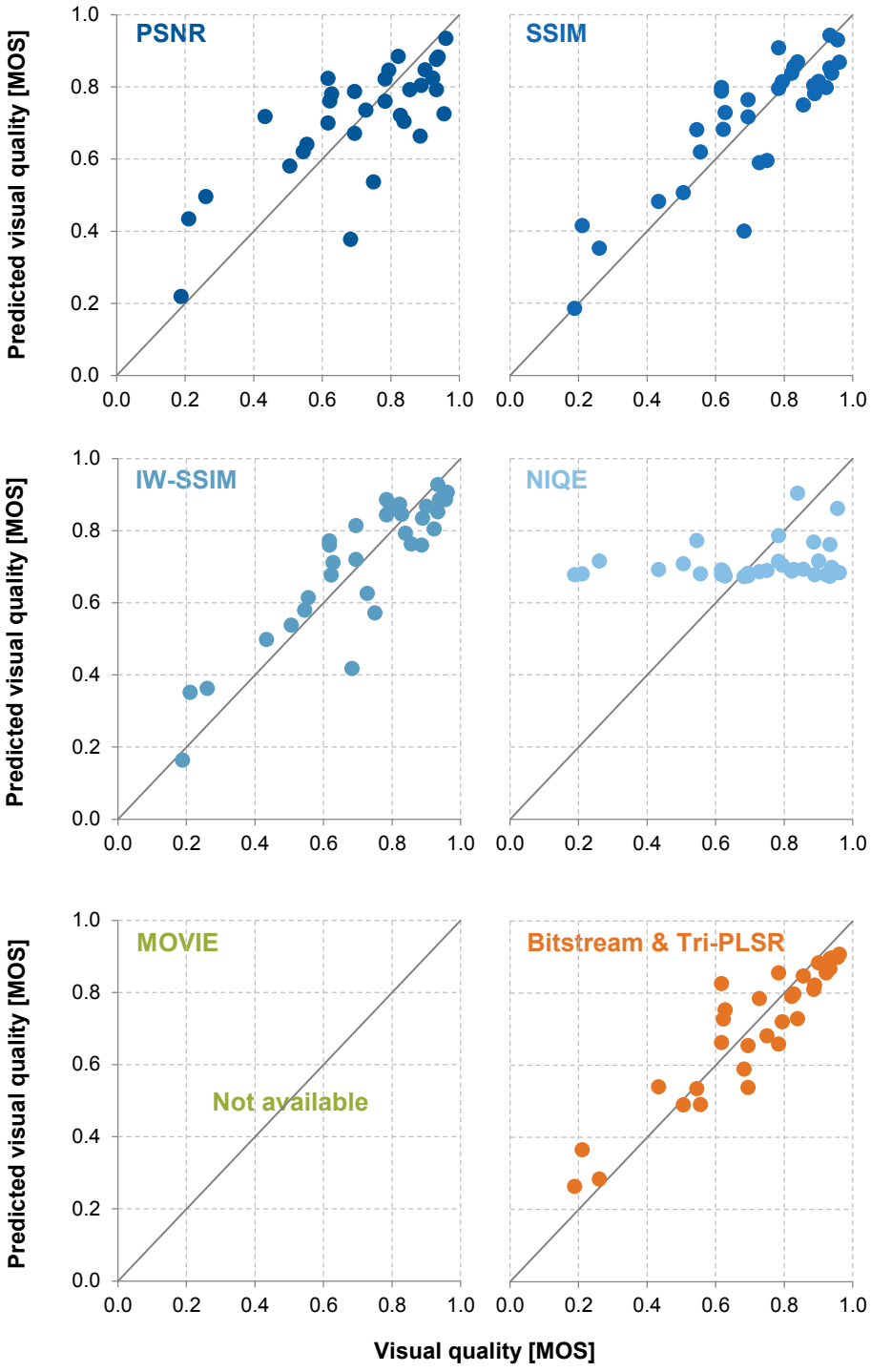

Figure B.23.: Scatter plots of the prediction results of the bitstream-based example metric with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE with cubic data fitting for the TUM1080p25 data set, showing the visual quality y against the predicted quality $\widehat{y}$

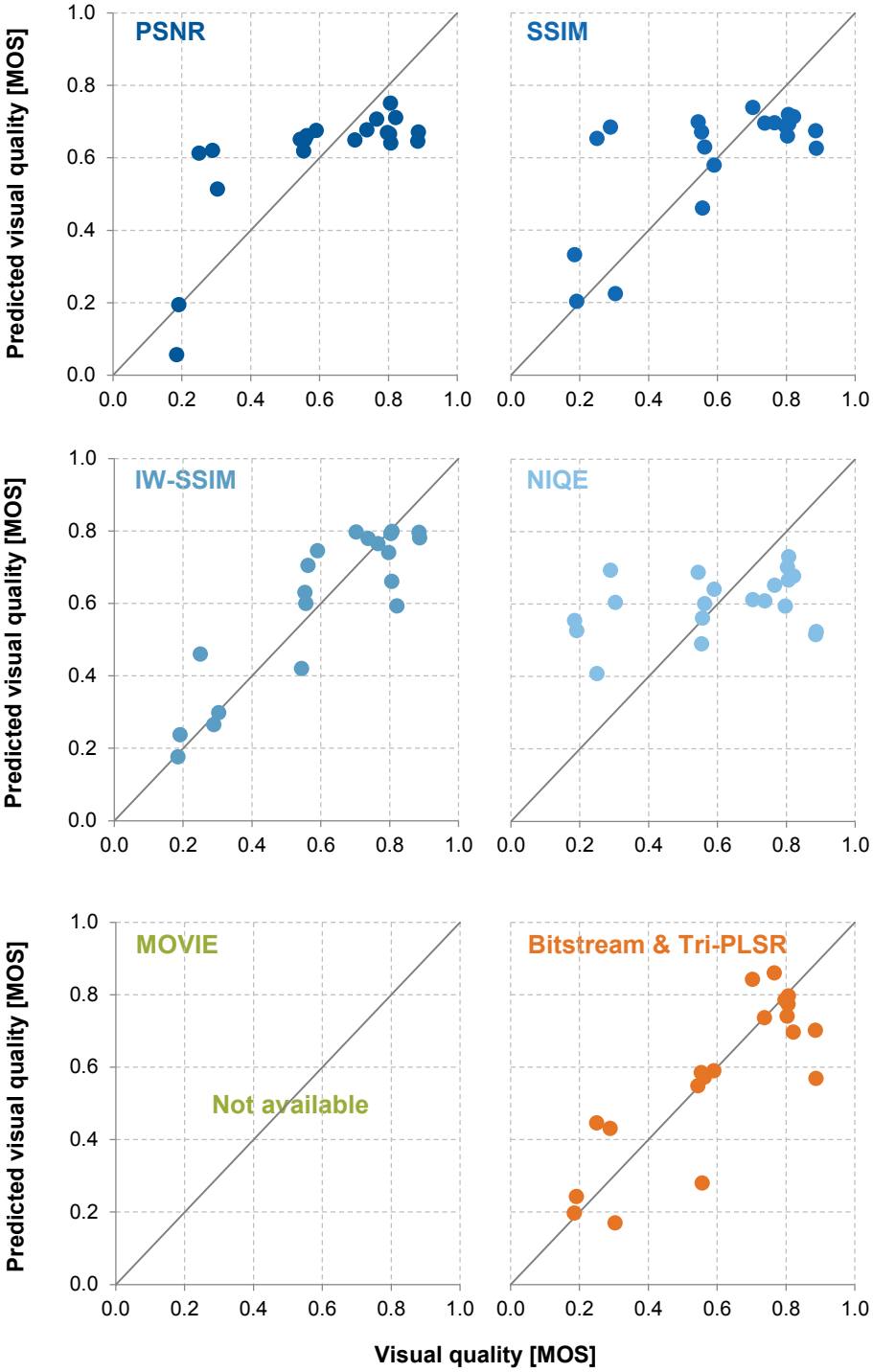

Figure B.24.: Scatter plots of the prediction results of the bitstream-based example metric with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE with cubic data fitting for the TUM1080p50 data set, showing the visual quality y against the predicted quality $\widehat{y}$

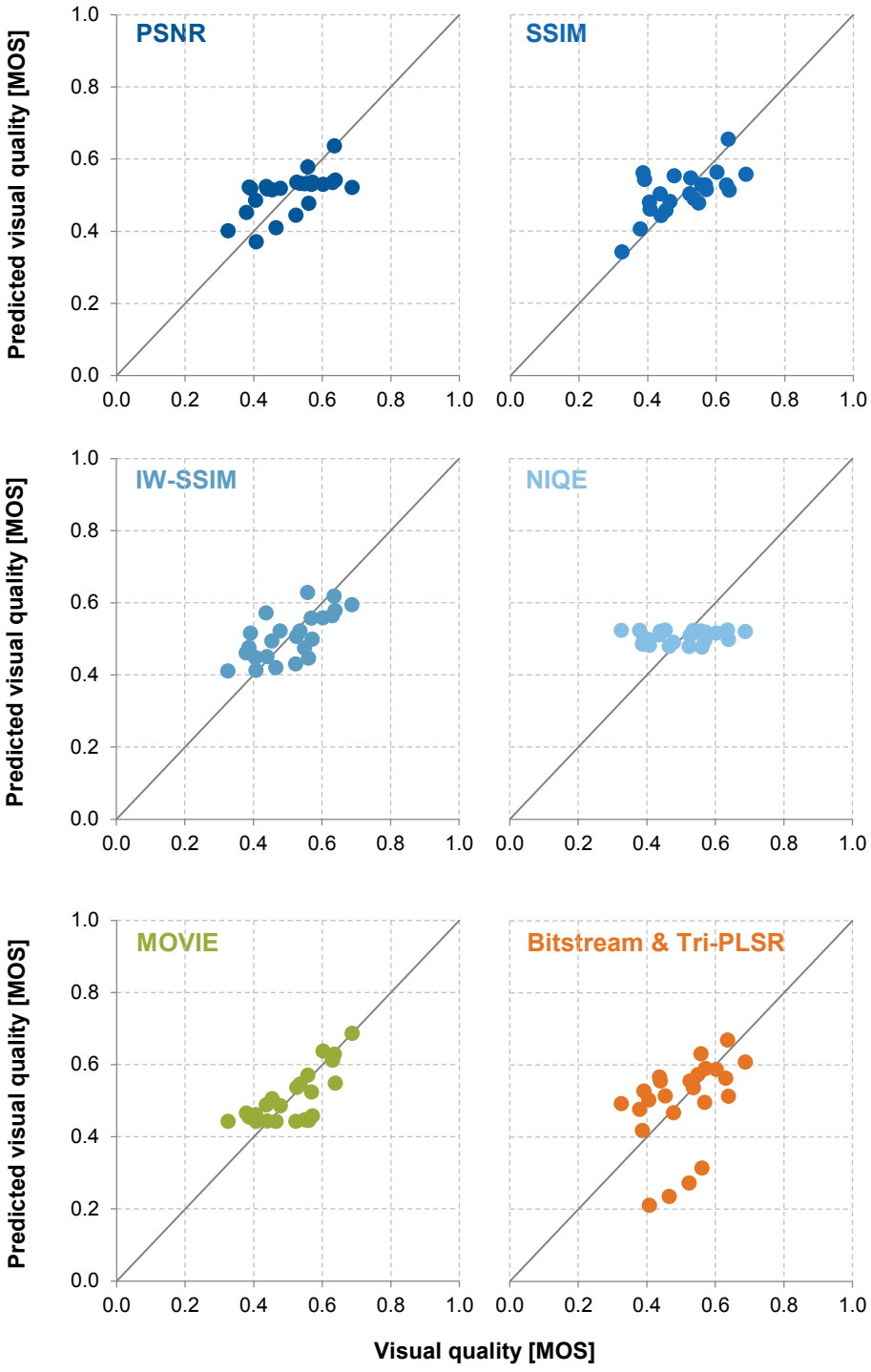

Figure B.25.: Scatter plots of the prediction results of the bitstream-based example metric with trilinear PLSR (Tri-PLSR) compared to PSNR, SSIM, IW-SSIM, NIQE and MOVIE with cubic data fitting for the LIVE data set, showing the visual quality y against the predicted quality $\widehat{y}$

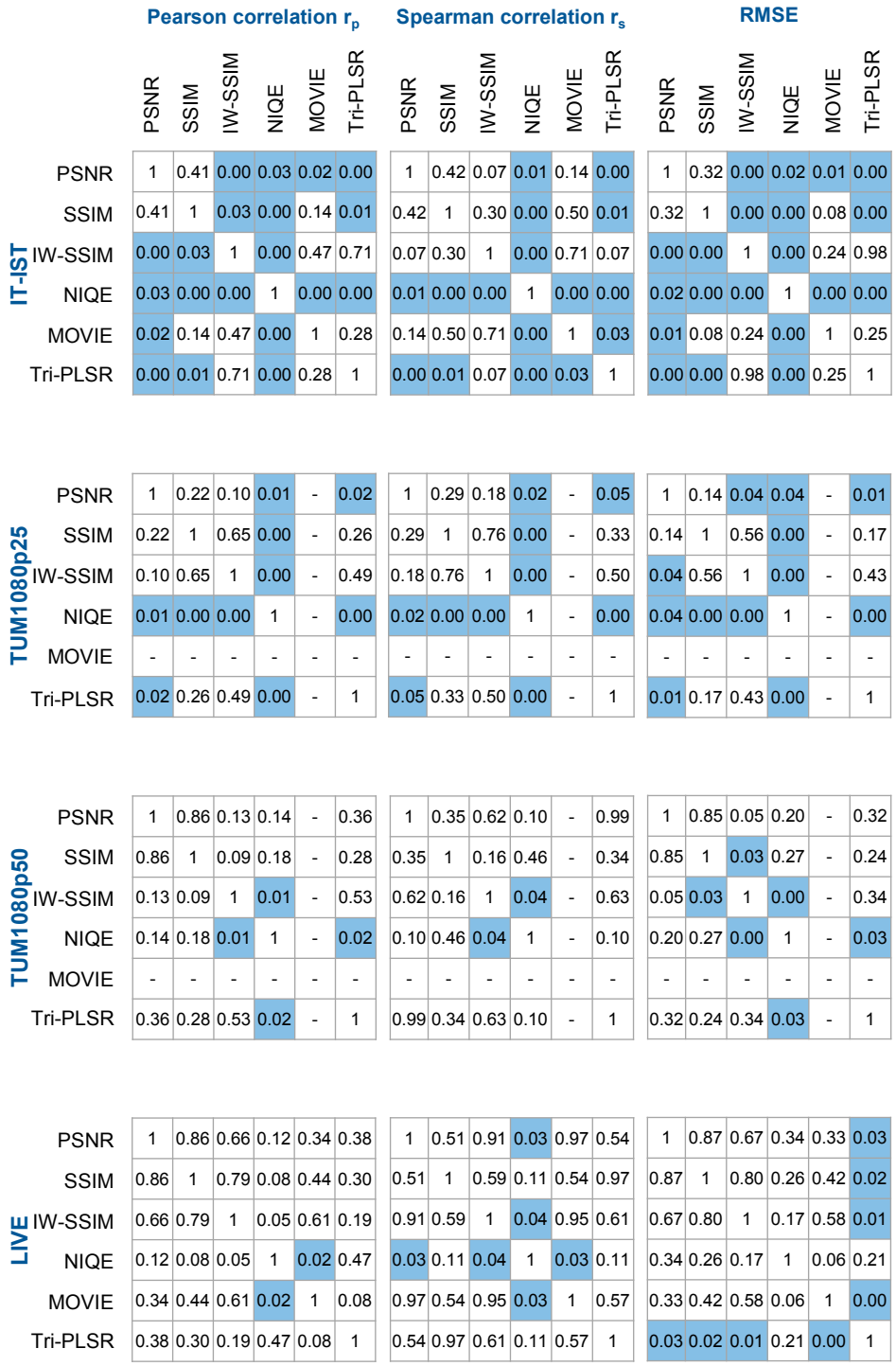

Figure B.26.: Statistical significance of the difference between prediction results of selected metrics with cubic data fitting and the bitstream-based example metric. For each combination the p-value is provided and results that are statistical significant at the 0.05 level with $p < 0.05$ are highlighted

List of Algorithms

Nomenclature and Symbols

$\circledast$	Hadamard product
$\odot$	Khatri-Rao product
$\otimes$	Kronecker product
$\overline{\bullet}$	average of variable
$\langle \bullet, \bullet \rangle$	inner/scalar product
$\bullet^{-1}$	inverse
$\bullet^{+}$	Moore-Penrose pseudoinverse
$\times_n$	mode-n product
$\| \bullet \|$	(Frobenius) norm of multi-way array
$\mathbf{1}$	vector of ones
$\circ$	outer product
$\widehat{\bullet}$	prediction/estimation of variable
$\operatorname{rank} \bullet$	rank
$\operatorname{rank}_n \bullet$	n-rank
$\bullet^{\top}$	transpose operator
$\operatorname{vec} \bullet$	vec operator
i, j, k	running indices for mode-1, mode-2 and mode-3
I, J, K	maximum of running indices i, j, k
a, b, c	scalar; element of multi-way array
$\mathbf{a}, \mathbf{b}, \mathbf{c}$	vector, fibre, one-way array
$\mathbf{A}, \mathbf{B}, \mathbf{C}$	matrix, two-way array

$\underline{\mathbf{A}}, \underline{\mathbf{B}}, \underline{\mathbf{C}}$ . . . . . multi-way array, usually three-way array

$\mathbf{A}_i$ . . . . . i-th slice of $\underline{\mathbf{A}}$

$\mathbf{A}_{I \times JK}$ . . . . . unfolded three-way array; here mode-1 unfolded

$\mathbf{e}, \mathbf{E}, \underline{\mathbf{E}}$ . . . . . error vector, two-way and three-way array

$\widehat{\mathbf{b}}, \widehat{\mathbf{B}}$ . . . . . prediction weights for features $\mathbf{x}_u$, $\mathbf{X}_u$

$\widehat{\mathbf{b}}, \widehat{\mathbf{B}}$ . . . . . prediction weights for not preprocessed features $\mathbf{x}_u$, $\mathbf{X}_u$

$\underline{\mathbf{G}}$ . . . . . core array

$\mathbf{I}$ . . . . . identity matrix

$\underline{\mathbf{I}}$ . . . . . identity multi-way array

m . . . . . feature index; representing mode-2

M . . . . . number of features; maximum value of m

n . . . . . video sequences index; representing mode-1

N . . . . . number of video sequences; maximum value of n

$\mathbf{p}$ . . . . . loadings vector for one component

$\mathbf{P}$ . . . . . loadings matrix for multiple components

$\widehat{\mathbf{q}}, \widehat{\mathbf{Q}}$ . . . . . prediction weights for features represented by scores $\mathbf{t}_u$

r . . . . . running index for components

R . . . . . number of used components

r_k . . . . . Kendall rank order correlation coefficient

r_p . . . . . Pearson correlation coefficient

r_s . . . . . Spearman rank order correlation coefficient

$\underline{\mathbf{S}}$ . . . . . video sequence

$\mathbf{S}_t$ . . . . . t-th frame of video sequence

s_{uv} . . . . . pixel in frame of video sequence

$\mathcal{S}$ . . . . . set of all video sequences $\underline{\mathbf{S}}$

$\mathcal{S}_C$ . . . . . calibration set

$\mathcal{S}_V$ . . . . . validation set

t . . . . . frame index; representing mode-3

T . . . . . number of frames in video sequence; maximum value of t

$\mathbf{t}$ . . . . . scores vector for one component

$\mathbf{T}$ . . . . . scores matrix for multiple components

u, v . . . . . pixel index in frame $\mathbf{S}_t$

U, V . . . . . pixels in frame $\mathbf{S}_t$

$\mathbf{w}, \mathbf{W}$ . . . . . weights with respect to optimisation criterion

$\mathbf{w}^M, \mathbf{w}^T$ . . . . . weights with respect to optimisation criterion per mode

x_m . . . . . m-th feature

$\mathbf{X}$ . . . . . two-way feature array (of pooled features)

$\mathbf{x}, \mathbf{x}_t$ . . . . . feature vector, feature vector of t-th frame

$\mathbf{X}_t$ . . . . . feature slice of t-th frame for n video sequences

$\underline{\mathbf{X}}$ . . . . . three-way feature array

$\widehat{\mathbf{X}}, \widehat{\underline{\mathbf{X}}}$ . . . . . approximation of $\mathbf{X}, \underline{\mathbf{X}}$

$\widetilde{\mathbf{y}}, \widetilde{\mathbf{X}}$ . . . . . residual of $\mathbf{y}, \mathbf{X}$

y . . . . . visual quality of n-th video sequence

$\mathbf{y}$ . . . . . visual quality of all N video sequences

$\widehat{y}$ . . . . . visual quality prediction of n-th video sequence

$\widehat{\mathbf{y}}$ . . . . . visual quality prediction of all N video sequences

Acronyms

2D-PCR	2D Principal Component Regression
ACR	Absolute Category Rating
ACR-HR	Absolute Category Rating - Hidden Reference
ALS	Alternating Least Squares
AVC	Advanced Video Codec (in H.264/AVC)
BTC	Basic Test Cell
CIE	Commission internationale de l'éclairage
CRT	Cathode Ray Tube
DCR	Degradation Category Rating
DCT	Discrete Cosine Transformation
DMOS	Differential Mean Opinion Score
DSCQS	Double Stimulus Continuous Quality Scale
DSIS	Double Stimulus Impairment Scale
DSUR	Double Stimulus Unknown Reference
fps	frames per second
HDTV	High Definition TV
HVS	Human Visual System
IEC	International Electrotechnical Commission

ISO . . . . . International Organization for Standardization

ITU . . . . . International Telecommunication Union

ITU-R . . . . . ITU Radiocommunication Sector

ITU-T . . . . . ITU Telecommunication Standardization Sector

JM . . . . . Joint Model

JND . . . . . Just Noticeable Difference

LCD . . . . . Liquid-Crystal Display

LVMR . . . . . Latent Variable Multivariate Regression

MLR . . . . . Multiple Linear Regression

MOS . . . . . Mean Opinion Score

MPEG . . . . . Motion Pictures Expert Group

MSE . . . . . Mean Squared Error

MSEC . . . . . Mean Square Error of Calibration

MSECV . . . . . Mean Square Error of Cross Validation

MSEP . . . . . Mean Square Error of Prediction

N-PLS . . . . . N-way Partial Least Squares

NIPALS . . . . . Nonlinear Iterative Partial Least Squares

NSS . . . . . Natural Scene Statistics

PARAFAC . . . . . Parallel factors

PCA . . . . . Principal Component Analysis

PCR . . . . . Principal Component Regression

PLS . . . . . Partial Least Squares

PLSR . . . . . Partial Least Squares Regression

PRESS . . . . . Prediction Sum of Squares

PSNR . . . . . Peak Signal to Noise Ratio

QoE . . . . . Quality of Experience

QoP . . . . . Quality of Perception

QoS . . . . . Quality of Service

RMSE . . . . . Root Mean Squared Error

RMSEC . . . . . Root Mean Square Error of Calibration

$RMSE_E$ . . . . . Epsilon-insensitive Root Mean Squared Error

RMSEP . . . . . Root Mean Square Error of Prediction

SAMVIQ . . . . . Subjective Assessment of Multimedia VIdeo Quality

SDSCE . . . . . Simultaneous Double Stimulus Continuous quality Evaluation

SDTV . . . . . Standard Definition TV

SSCQE . . . . . Single Stimulus Continuous Quality Evaluation

SSIM . . . . . Structural Similarity Index

SSIS . . . . . Single Stimulus Impairment Scale

SSMM . . . . . Single Stimulus MultiMedia

SSMR . . . . . Single Stimulus with Multiple Repetitions

SSNCS . . . . . Single Stimulus Numerical Categorical Scale

SVC . . . . . Scalable Video Codec

SVD . . . . . Single Value Decomposition

SVM . . . . . State Vector Machine

VQEG . . . . . Video Quality Experts Group

Index