ON THE ACCURACY OF SHORT-TERM QUALITY MODELS FOR LONG-TERM QUALITY PREDICTION

Marie Neige Garcia¹, Werner Robitza², Alexander Raake¹

¹Technische Universität Berlin, ²Deutsche Telekom AG, Berlin, Germany

ABSTRACT

With video services such as HTTP-based adaptive streaming, network congestion may result in quality fluctuations over several minutes. There is therefore a need for estimating the quality of long audiovisual sequences. This can be achieved by using short-term audiovisual quality models, which output quality scores for short periods of time, for instance 10 s. Temporal pooling such as averaging is typically applied on the short-term quality estimates for providing a quality score for a longer time period, for instance three minutes. With this modeling strategy, the performance of the overall quality model can be increased by improving both the short-term quality model and the temporal pooling strategy. However, depending on the temporal pooling strategy, and possibly the targeted test data obtained for long sequences, a small improvement of the short-term quality model may eventually not have any significant impact on the long-term quality estimates. This paper investigates this aspect by comparing the performance results of the combination of six short-term quality models with six different pooling strategies. Results show that the performance of well performing short term models is a good indicator of the performance of the long-term quality models, independently of the pooling strategy.

Index Terms— Quality of Experience, audiovisual quality, video quality, adaptive streaming, temporal pooling.

1. INTRODUCTION

Audio, video, and audiovisual quality metrics are typically designed for short-term audiovisual sequences, see ITU-T Rec. G.1011. They estimate quality scores for periods of approximately 10 s. This time restriction is closely linked to the design of the subjective tests used for training and/or validating the metrics. Indeed, these subjective tests make use of short sequences (e.g., ITU-T BT-500.13), thus limiting the influence of the location of the distortions in the short sequence and allowing the use of a large amount of contents and test conditions without tiring the test participants.

The use of short test sequences is appropriate to assess the quality impact due to homogeneous distortions or due to relatively short distortions compared to the overall sequence duration. However, with increasingly-used services such as HTTP-based adaptive streaming (HAS), distortions occur over longer periods, from a few seconds to several minutes.

Indeed, for HAS services, servers provide the contents in segments of typically 2 to 10 seconds (ISO/IEC 23009-1), encoded at different audio and/or video bitrates. Segments are available in different quality levels – also called representations –, forming the adaptation set. Representations are typically characterized by audio and video bitrate, video frame-rate, resolution and/or encoding quantization settings. The client can dynamically adapt the choice of segment representation to match the available throughput. Switching between segments may result in quality fluctuations over periods from the segment duration (a few seconds) to several minutes, in case of long periods of the same representation. Depending on the efficiency of the adaptation strategy and the available throughput, stalling events of few to several seconds may occur.

Models for estimating the quality of long sequences hence require short-term (ST) quality estimation and long-term (LT) integration strategies. In this paper, “long-term” corresponds to a time span of up to three minutes, and not tens of minutes or even years, as defined in other works, e.g. [1].

In the audio/speech domain, several attempts have been made for estimating long-term (e.g. 3 min) quality from ST subjective quality scores [2, 3]. Major findings are the higher impact of low quality periods and late distortions on the overall perceived quality, also referred to as peak-end effect [4]. The peak effect was reported in [5] to be stronger than the end effect for audio-visual conversation scenarios.

For video, the ST quality scores are generally estimated per frame [6, 7], per (e.g., DASH-) segment or per period of constant quality levels [8]. ST quality scores are typically estimated using full-reference (FR) metrics (PSNR, SSIM) [7]. Alternatively, encoding parameters such as the encoding bitrate or the quantization parameters [6], the quality levels, or the subjective scores [7] can be used. The ST quality scores are then pooled over time to provide a quality score for the whole sequence. Pooling strategies are for instance averaging, or Minkowski summation, which are typically used in the video domain to obtain estimates of 10 s quality scores from per-frame quality scores.

An interesting comparison of thirteen pooling strategies has been reported in [7], showing the good performance of
a pure averaging of PSNR and SSIM scores compared to the other pooling methods. However, they do not extensively analyze the impact of the performance of the ST quality models on the performance of the LT models and on the influence of the pooling strategy. These points are crucial for the development of useful LT audiovisual quality models. We are investigating them in this paper and in the context of HAS services. The investigation methodology is presented in Section 2. Subjective test databases used for the analysis are described in Section 3. ST quality models and LT pooling strategies are introduced and combined in Section 4. In total, there are 36 combinations of ST and LT pooling models. Their performance on the subjective test databases is analyzed in Section 5 before concluding.

2. METHODOLOGY

In the following, we assume that LT quality scores are estimated by temporally pooling quality estimates of ST quality models. With this assumption, we want to address the following research questions: If an ST model is performing better than other models on ST subjective quality database(s), will it yield better estimates of the LT quality scores? Does it depend on the temporal pooling strategy?

To answer these questions, we will make use of three subjective test databases: ST1, LT1 and LT2. ST1 is a short-term (10 s) video quality database. ST1 addresses the quality-impact due to encoding video bitrate (CBR), frame-rates, resolution, and Group-Of-Picture (GOP) structure characteristics, see Sect. 3.1. LT1&2 are long-term (LT1: 42 - 71 s, LT2: 3 min) audiovisual quality databases developed for the quality assessment of HAS services. Degradations are quality switches between different representations and stalling events, see Sect.3.2.

Encoding settings, bitrate ranges and encoding resolution ranges are similar between all databases. This is crucial for our analysis. It ensures that subjective quality scores obtained in ST1 are a good approximation of the subjective quality scores we would obtain for each encoded segment of LT1&2 if these segments had been presented to test participants in an ST subjective test. This should also ensure that an ST model trained on ST1 yields good estimates for each segment of a long sequence.

To address our research questions, we will first compute the performance of five ST video quality models on ST1. We will then apply the same temporal pooling strategies for all models and compute performance results on LT1&2. Models will be ranked based on their performance. We will check whether the ranking order of the models is maintained between ST1 and LT1&2, and if models that are significantly different on ST1 remain significantly different on LT1.

As performance indicator, we selected the root-mean-squared-error (RMSE), which captures the accuracy of the model. Lower RMSE values indicate better model performance. In this paper, quality scores used for computing the RMSEs are expressed on the 5-point quality scale [1,5]. We also check whether the difference of RMSE between models is statistically significant, as described in ITU-T P.1401.

3. SUBJECTIVE TEST DATABASES

3.1. Short-term video quality test (ST1)

In order to evaluate the ST quality models, we have conducted a video-only subjective quality test. Eight High Definition (HD, 1920x1080 pixels) source contents (SRCs) were used. They differ in terms of spatio-temporal complexity and are representative of content types for video streaming services (movies, movie trailers, music videos, documentaries). They were encoded using ffmpeg with the x264 encoder at High profile, at different bitrates, frame-rates and resolutions, with constant bitrate mode (CBR), and two-pass encoding. The preset setting for x264 was medium, except for one test condition in which it was set to superfast. No scene cut detection was applied, except in one test condition. We have also varied the Group Of Picture (GOP) length and structure. Overall, 21 test conditions were applied to each SRC, resulting in 168 processed video sequences (PVS). See Table 1 for more details.

3.2. Long-term audiovisual quality tests (LT1, LT2)

In order to evaluate the LT quality models, we conducted two LT audiovisual quality tests, LT1 and LT2. LT1&2 use 42- to-71 s and 3 min-long sequences, respectively. 66 SRCs were used in LT1, and 22 in LT2. Parts of the SRCs were uncompressed HD (1920x1080 pixels). They were downloaded from the Consumer Digital Video Library (www.cdvl.org), from the Blender open projects (www.blender.org), or were bought by our laboratory for internal use and research purposes. The rest of the SRCs were originally 4K videos and were acquired from online video streaming services. These 4K videos were downsampled to HD before applying any encoding. Each SRC was perceptually checked for quality. The number of bits per pixel, bpp, was computed as additional quality indicator, and videos with high bpp were preferably selected.

22 test conditions were used in LT1. Each condition was applied on 3 SRCs, resulting in 66 test sequences (PVS).
For LT2, 11 test conditions were applied on 2 SRCs, resulting in 22 PVSs. For both tests, SRCs were not repeated to keep test subjects entertained and make the tests more realistic. Special care was taken when assigning SRCs to test conditions. In particular, if quality switches were not visible due to low spatio-temporal complexity of the SRCs, this SRC was replaced by a more complex one.

As described in Table 2, three representations (QLs: Quality Levels) were used in both tests. In LT1, the representations differ only in terms of video encoding bitrates, while in LT2, the video resolution and audio and video bitrates (CBR) were varied. In LT1, there were three test conditions with constant quality (low, medium, high), 12 with quality switches (ten with drops and two with ramp-up), two with stalling only, and five with both stalling and quality switches (drops). To avoid a strong recency effect, which was not the focus of LT1, no quality degradation was applied in the last 15 s of the sequences.

In LT2, there were two test conditions with constant quality (low, high), five with quality switches (three with drops, two with peaks), one with stalling degradation only, and three with both stalling and quality switches (two with drops, one with ramp-down).

<table>
<thead>
<tr>
<th>Test</th>
<th>Test conditions</th>
</tr>
</thead>
</table>
| **LT1** | 3 QLs:(1080,8500,128),(1080,1500,128), (1080,500,128)  
HE-AACv2, x264 high profile  
QS:(drops,abrupt,[5,15,25] s,#{1,3,5},{1,2})  
(ramp-up,gradual,[10,40] s,#{1})  
L:10 s, N: 1 |
| **LT2** | 4 QLs:(1080,10000,196),(1080,2500,128), (480,500,96),(240,500,196)  
AAC-LC, x264 high profile  
QS:(drops,gradual,[30,105] s,#{1,3,5},2)  
(peaks,gradual,[30,105] s,#{1,3,5},2)  
L:12-15 s, N: [1,2] |

Table 2. LT1&2 test conditions. QLs: quality levels (encoding resolution (width in pixels), encoding video bitrate (kbps), encoding audio bitrate (kbps)); QS: Quality Switch description (type: [drops, ramp-up, peaks], speed: [abrupt, gradual], duration in sec, #: amount, depth (number of QLs)); L,N: duration (s) and number of stalling events.

To generate the test sequences, the SRCs were first encoded at each quality level using ffmpeg and split into segments of 5 s length. These segments were then recombined according to the quality fluctuation patterns described in the test conditions. For test conditions with stalling, a spinning rebuffering indicator was inserted.

3.3. Test method and environment

The Absolute Category Rating (ACR) test method of ITU-T P.910 and P.911 was used in all tests with a five-point categorical quality scale. For ST1, stimuli were randomized, with the constraint that two consecutive stimuli do not correspond to the same SRC. All subjects viewed all stimuli, but in different presentation orders.

30 subjects participated in each test. They were screened for visual acuity and color blindness. Participants started each test with a training phase in order to adapt to the rating tool as well as to the content types, quality range and degradation types used in the test. Subjects were asked to judge the video quality (ST1) or the overall audiovisual quality (LT1&2). If the correlation between the ratings of each subject and the ratings averaged over all subjects was smaller than 0.7 for a subject, ratings for that subject were discarded. Subsequently, the ratings were averaged per PVS over all subjects, giving a mean opinion score per PVS, MOSPVS.

Video was displayed on a professional 42 inch display while audio was presented via high-quality headphones. Devices were calibrated according to ITU-T P911.

4. MODEL DESCRIPTION

In order to estimate the quality of LT audiovisual sequences, we use the ST-to-LT model structure shown in Fig 1. The ST models take information about segments of typically 2-to-10 s duration as input. Here, different ST models work with different types of information. For each segment, an ST model may use a coarse description of the segment, for instance its representation (audio and video bitrate, frame rate, resolution). Another ST model may use the full pixel info, or even a uni-modal quality estimate for that segment, obtained for instance with an FR model (e.g., VQM scores for video [9]). Apart from the dummy model (see Sect 4.1), the per-segment audio and video quality scores are sent to the audiovisual module to provide ST audiovisual (AV) quality scores (QST). These scores are integrated over time by the “Pooling” module to provide a long-term audiovisual quality estimate (QLT). In parallel, quality degradation due to stalling (degStal) is estimated by the stalling degradation module. The stalling degradation and per-sequence audiovisual quality estimates are combined to provide an integrated quality score for the whole sequence (Q).

Fig. 1. Model overview. From ST models to LT models.
4.1. Short-term audiovisual quality models

<table>
<thead>
<tr>
<th>ID</th>
<th>Video module (STvideo)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VQMVQ</td>
<td>general VQM model</td>
</tr>
<tr>
<td>PSNRVA</td>
<td>PSNR averaged per segment</td>
</tr>
<tr>
<td>DT0</td>
<td>same as DT0, QScalV retrained on ST1</td>
</tr>
<tr>
<td>DT2</td>
<td>bitstream-based, QScalV retrained on ST1</td>
</tr>
<tr>
<td>Dummy</td>
<td>5-point scale quality levels</td>
</tr>
</tbody>
</table>

Table 3. ST audiovisual quality models.

ST models are described in Table 3. It is noted that in this paper, the “Dummy” model is treated separately, see end of Sect. 4.1. For the other ST models, the ST audio quality and audiovisual integration modules are fixed, and only the ST video quality modules are interchanged. The ST audiovisual integration module is taken from the audiovisual quality module of ITU-T P.1201.2. This does not limit the generality of the research presented here, and enables a clear focus on the impact of ST video models on LT audiovisual quality prediction. The ST audiovisual quality scores are computed as follows:

\[
Q_{ST} = av_1 + av_2 \cdot Q\text{cod}_A + av_3 \cdot Q\text{ScalV} + av_4 \cdot Q\text{cod}_A \cdot Q\text{ScalV},
\]

where \(Q\text{cod}_A\) and \(Q\text{ScalV}\) are the audio and video coding distortions due to low audio and video encoding bitrates, and \(av_n, n \in [1, 4]\) are the model coefficients. \(Q_{ST}\) is expressed on a 100-point scale \([0, 100]\) and can be transformed to the traditional 5-point quality ACR scale \([1, 5]\) using the \(MOS_{frQ}\) function in Sect 6.4 of ITU-T P.1201.2.

The audio-quality term \((Q\text{cod}_A)\) of \(Q_{ST}\) is computed using the audio coding module of ITU-T P.1201.2 [10]:

\[
Q\text{cod}_A = a_{1A} \cdot \exp(a_{2A} \cdot \text{audioBitrate}) + a_{3A},
\]

where \(\text{audioBitrate}\) is the audio bitrate in kbps, and \(a_{nA}, n \in [1, 3]\) are the model coefficients.

In this study, video distortions \(Q\text{ScalV}\) may be due to low video encoding bitrate or upscaling. “Upscaling” occurs when the video is encoded at a reduced resolution, then blown-up to the display resolution, yielding blur.

In our study, we compared several models for estimating the \(Q\text{ScalV}\) of \(Q_{ST}\). The first one is the NTIA general model (aka \(V\text{QM}\)), standardized in ITU-T J.144 [9]. The VQM scores are expressed on the \([0, 1]\) scale, where higher scores reflect higher distortions. On \(ST1\), we observed a linear association between the VQM scores and the \(MOS_{PSNR}\) values. As a consequence, the VQM scores were transformed onto the 5-point quality scale using the following linear transformation: \(MOS_{\text{ScalV}} = -5 \cdot \text{VQM} + 5\).

The degradation score \(Q\text{ScalV}\) is then obtained from \(MOS_{\text{ScalV}}\) using eq. (3):

\[
Q\text{ScalV} = 100 - R_{frQ}(MOS_{\text{ScalV}}). \tag{3}
\]

**RfromMOS** is described in Annex C of ITU-T G.107.

As additional video quality metric, we computed the PSNR averaged over all video frames \((PSNR_{avg})\). As for VQM, the PSNR scores were transformed onto the MOS scale using the \(MOS_{PSNR}\) scores of \(ST1\) as target values: \(MOS_{\text{ScalV}} = 0.15 \cdot PSNR_{avg} - 2.15\). \(Q\text{ScalV}\) is obtained from \(MOS_{\text{ScalV}}\) using eq. (3).

We also used an extended version of the video coding module of ITU-T P.1201.2 [10]. This variant, \(DT0\), is expressed as follows:

\[
Q\text{ScalV} = Q\text{codV} + Q\text{ScalV}, \tag{4}
\]

\[
Q\text{codV} = a_{1V} \cdot \exp(a_{2V} \cdot \text{bpp}) + a_{3V} \cdot \text{Complexity} + a_{4V}, \tag{5}
\]

where \(\text{bpp}\) is the number of bits per pixel, \(\text{Complexity}\) estimates the spatio-temporal complexity of the error-free encoded content using I-frame sizes, and \(a_{nV}, n \in [1, 4]\) are the model coefficients.

\(Q\text{ScalV}\) is modeled using subjective test results from [11, 12]. It is expressed as follows:

\[
Q\text{ScalV} = 72.61 \cdot \log_{10}(0.32 \cdot (\text{scaling} - 1.0) + 1.0), \tag{6}
\]

\[
\text{with scaling} = \frac{\text{displayResolution}}{\text{encodingResolution}}, \tag{7}
\]

where \(\text{displayResolution}\) and \(\text{encodingResolution}\) are the number of pixels per frame for the display and for the encoded video.

We use two additional variants of \(DT0\): \(DT1\) and \(DT2\). For both variants, we retrained the upscaling module \(Q\text{scalV}\) on the \(ST1\) database. For \(DT2\), we also modify \(Q\text{codV}\) by using a bitstream-based parameter, \(QP\), instead of the frame-based \(\text{Complexity}\) parameter. \(QP\) corresponds to the normalized quantization parameters per Macroblock, averaged over all non I-frames. Quantization parameters are normalized by dividing them by the maximum value they can take (51). The resulting \(Q\text{codV}\) module of \(DT2\) is expressed as:

\[
Q\text{codV} = b_{1V} \cdot \exp(b_{2V} \cdot \text{bpp}) + b_{3V} \cdot QP + b_{4V} \cdot QP^2 + b_{5V}, \tag{8}
\]

Finally, we use a \(Dummy\) ST audiovisual quality model for which the encoding audiovisual quality \(Q_{ST}\) is derived from the number of quality levels (representations) in the adaptation set, as expressed in eq. (9):

\[
Q_{ST} = 1 + (QL - 1) \cdot \frac{5 - 1}{QP_{max} - QP_{min}}. \tag{9}
\]

\(QL\) is an integer representing the quality level. Higher \(QL\) values correspond to higher quality, and \(QP_{max}\) and \(QP_{min}\) are the highest and lowest quality levels, respectively. For instance, if there are four quality levels in the adaption set \((QL1, QL2, QL3, QL4)\), the respective quality scores will be 1, 2.33, 3.66 and 5.
4.2. Long-term quality integration

Long-term audiovisual quality is estimated using eq. (10):

\[ Q = Q_{LT} - degStal, \]  

with \( Q_{LT} = \text{pooling}(Q_{ST}) \), where \( Q_{ST} \) is the list of ST audiovisual quality scores for each segment of the processed test sequence. These quality scores are pooled over time using six different pooling strategies, see Table 4.

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>mean over all sequence</td>
</tr>
<tr>
<td>Perc90&amp;75</td>
<td>mean over 90%&amp;75% worst ST scores</td>
</tr>
<tr>
<td>Rosenbluth</td>
<td>peak-end effect [3]</td>
</tr>
<tr>
<td>WeissA&amp;B</td>
<td>peak-end effect [2], models A&amp;B</td>
</tr>
</tbody>
</table>

Table 4. Six pooling strategies.

By using \( \text{Perc75} \) and \( \text{Perc90} \), we investigate whether given more weights to low quality scores improves the pooling. \( \text{Rosenbluth} \) and \( \text{Weiss} \) models correspond to different weighted averages of the ST scores. They model the peak-end effect [4] with different recency durations and weights for the low quality ST segments.

The quality degradation due to stalling, \( \text{degStal} \), is computed using eq. (III-1) of the P.1201 Amd.2 and is reproduced in eq. (11):

\[ \text{degStal} = s_4 + s_1 \cdot \exp((s_2 \cdot L + s_3) \cdot N) \]  

The final quality score is obtained by subtracting \( \text{degStal} \) from the ST quality scores, see eq. (10).

5. MODEL PERFORMANCE COMPARISON

5.1. Short-term model performance comparison

We first compute the performance of ST models on ST1, see Table 5. The \( \text{RMSE} \) is computed between the \( \text{MOS}_{PVSs} \) and the estimated video quality for each PVS. Since the frame rate was not varied in LT1&2, the \( \text{RMSE} \) was computed only using PVSs without frame-rate reduction, that is, based on 112 sequences. Note that to compensate for bias in the subjective tests, the estimated quality scores are linearly mapped (\( \text{RMSE}_{lm} \)) to the subjective scores for each ST model. The RMSE between the subjective scores and the mapped scores are also provided (\( \text{RMSE}_{lm} \)).

The bitstream-based model \( \text{DT2} \) obtains the best performance results (\( \text{RMSE}_{lm} = 0.37 \)), followed by the FR model \( \text{VQM}_{AV} \) and the frame-based models \( \text{DT1} \) and \( \text{DT0} \). \( \text{PSNR}_{AVG} \) is significantly worse than all other models and, after linear mapping, \( \text{DT2} \) is significantly better than all other models. Otherwise, the difference between model performances is not significant (ITU-T P.1401).

Note the high difference between \( \text{RMSE} \) and \( \text{RMSE}_{lm} \) for \( \text{DT0} \). This difference is due to the upscaling module of \( \text{DT0} \), which is underestimating the degradation due to upscaling. This bias is compensated by the linear mapping. However, we will see in the Section 5.2 that this bias is highly impacting the performance of the LT models.

### Table 5

<table>
<thead>
<tr>
<th>ID</th>
<th>RMSE</th>
<th>( \text{RMSE}_{lm} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT2</td>
<td>0.40</td>
<td>0.37</td>
</tr>
<tr>
<td>( \text{VQM}_{AV} )</td>
<td>0.44</td>
<td>0.44</td>
</tr>
<tr>
<td>DT1</td>
<td>0.47</td>
<td>0.44</td>
</tr>
<tr>
<td>DT0</td>
<td>0.57</td>
<td>0.45</td>
</tr>
<tr>
<td>( \text{PSNR}_{AV} )</td>
<td>0.68</td>
<td>0.67</td>
</tr>
<tr>
<td>Dummy</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

Fig. 2. RMSE values for all ST models and pooling strategies.

The overall best performing LT model is the \( \text{DT2} \) model with \( \text{Perc90} \) pooling (\( \text{DT2} + \text{Perc90}, \text{RMSE}_{lm} = 0.43 \)). It is performing significantly better than all models for which the \( \text{RMSE} \) value is above the dashed black line in Fig. 2.

Note that the dummy+\( \text{Perc75} \) model is not performing significantly worse than the \( \text{DT2}+\text{Perc90} \) model. However, we can expect from their \( \text{RMSE}_{lm} \) values (dummy+\( \text{Perc75} \): 0.51, \( \text{DT2}+\text{Perc90} \): 0.43) that the performance difference would become significant with a higher number of PVSs.

It is also remarkable that the dummy model is performing better than both \( \text{PSNR}_{AV} \) and \( \text{DT0} \). The dummy model seems to be an interesting anchor for detecting ST models.
with strong bias.

It can be observed in Fig. 2 that the impact of the ST model on the overall quality is much higher than the impact of the pooling method. In particular, the ranking of the best performing ST models in Table 5 is globally maintained in Fig. 2, although it slightly depends on the pooling strategy. The ranking between $PSNR_{AV}$ and $DT_0$ models is not maintained. It seems that the bias introduced by the upsampling module of $DT_0$ is impacting all pooling methods, more than the bad predictions of the $PSNR$ metric. Overall, it seems that the ranking of the performance results of ST models which are well performing on the ST database is a good indicator of the performance of the corresponding LT models, independently of the pooling strategy. This observation is valid for well-performing ST models, not for ST models with strong bias, that is, where high differences are observed between $RMSE$ and $RMSE_{lm}$.

For a given ST model, there is no significant difference of $RMSE_{lm}$ values between the different pooling strategies. Although the differences of $RMSE$ values are generally not significant between different pooling strategies, there is a tendency, especially for the best performing ST models, that the $AVG$ and $Perc90$ pooling both yield better performance results than the other pooling strategies.

A more detailed analysis of the $RMSE_{lm}$ values show that the ranking of the pooling strategies in terms of $RMSE$ depends on the LT databases. The $Weiss$, and $Rosenbluth$ pooling strategies tend to yield better performance results on $LT1$. Opposite results are observed on $LT2$, where $AVG$ and $Perc90$ tend to yield the best performance.

Finally, we found that for all pooling strategies, $DT2$ is performing better than $DT1$, and both $DT2$ and $DT1$ are performing significantly better than $DT0$. As previously mentioned, the rank order of the ST models is maintained when computing performance results of LT models. This also shows that when two ST models have significantly different $RMSE$ values, the corresponding $RMSE_{lm}$ values are also significantly different. These results should of course be taken with care since they depend on the characteristics and number of sequences in the subjective test databases. However, it can be concluded that retraining ST models on $ST1$ had a direct positive impact on the performance of the LT models. Also really interesting is the close agreement between $RMSE$ values on $ST1$ for the three DT models ($DT2$: 0.4, $DT1$: 0.47, $DT0$: 0.57) and $RMSE_{lm}$ values of the corresponding LT-models, for instance using an average pooling strategy ($DT2$: 0.44, $DT1$: 0.47, $DT0$: 0.6).

### 6. CONCLUSION

This paper tackles the topic of long-term (LT) audiovisual quality prediction in the context of HTTP-based adaptive video streaming services, where “long-term” is in the magnitude of a few minutes. Assuming that LT quality scores are estimated by temporally pooling quality estimates of short-term (ST) quality models, we analyzed whether the performance of ST models is a good indicator of the performance of LT models. The analysis was done using 36 combinations of ST audiovisual quality models and pooling models. ST models differ only in terms of video quality estimates. Performance results of the ST models were computed on 112 10 s video sequences from a single subjective test database. Performance results of LT models were calculated on 88 42-to-3 min audiovisual sequences from two subjective test databases. We showed that the performance of the video module of the ST models on 10 s sequences is a good indicator of the performance of the LT models, independently of the pooling strategy. In particular, rank orders of well-performing ST models are maintained for the corresponding LT models. We also showed that improving ST models by retraining them on the ST database yields improvement on the LT databases, and that when these ST models were significantly different, the corresponding LT models were significantly different as well. To validate our results, additional LT databases with sequences of different quality fluctuation patterns are needed. With this increased number of LT databases, we will also investigate how much – in terms of reducing RMSE – we need to improve the best performing ST model to improve the corresponding LT model significantly.

### References


