A Modular HTTP Adaptive Streaming QoE Model – Candidate for ITU-T P.1203 ("P.NATS")

Werner Robitza*, Marie-Neige Garcia†, and Alexander Raake‡
*Telekom Innovation Laboratories, Deutsche Telekom AG, Germany
Email: werner.robitza@telekom.de
†Assessment of IP-based Applications, TU Berlin, Germany
Email: marie-neige.garcia@tu-berlin.de
‡Audiovisual Technology Group, TU Ilmenau, Germany
Email: alexander.raake@tu-ilmenau.de

***Abstract*—This paper describes a quality model for HTTP Adaptive Streaming. It integrates existing audio and video quality scores to a final quality estimation, factoring in quality variations over time, the recency effect, as well as location and length of buffering events at the player side. We built the model based on data gathered from more than 17 subjective quality tests. It was submitted to the ITU-T P.NATS competition; parts of it have since been released in the official recommendation ITU-T P.1203.3 as an "audiovisual quality integration module". In the context of standardization, the model was validated on 30 subjective databases, showing high performance. Its modular approach allows its components to be re-used in other applications and combined with different temporal pooling techniques.**

Keywords*—*Video quality; Quality of Experience; Quality model; HTTP Adaptive Streaming, Standardization

I. Introduction

HTTP adaptive streaming (HAS) has become the predominant method of delivering video to end users via the Web, often replacing traditionally employed streaming methods (e.g., requiring the RTP protocol and setting-up of dedicated networks). HAS dynamically adapts the video to account for changes of effective throughput and other data delivery problems. It does so by switching between video representations (i.e., streams encoded at different bitrates and/or resolutions), effectively minimizing the required bandwidth while retaining playback continuity. This reduces the chance of buffer exhaustion, which in turn would lead to annoying stalling artifacts. However, the switching itself can also affect the user-perceived audiovisual quality.

At this stage, Quality of Experience (QoE) for HAS has already been studied for years by the academic and industrial communities. We know about the factors influencing overall enjoyment and frustration of streaming users. *Quantifying* these impacts however is a challenge in itself: it requires the creation of large subjective test databases or the monitoring of streams in real life, both with their advantages and trade-offs. While (standardized) instrumental video quality models have existed for years, they mostly predict quality for short sequences only (e.g., 10 s), not taking into account the quality variation and stalling effects typical for HAS.

A standardized way to measure HAS QoE is generally desirable. It allows for better comparison of existing market deployments and enables a common, quantifiable understanding of QoE. For this reason, the International Telecommunication Union's Study Group 12 (SG12), Question 14 (Q14), pursued the development of an HAS QoE model, in a competition called "P.NATS" (*Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport*). This paper describes a model we created as a candidate for P.NATS—more specifically, a module that integrates existing short-term video and audio quality scores. Our model consists of independent components that cover different aspects of subjective quality perception. It works in a video or audio codec-agnostic fashion and has been designed for sequences up to five minutes length. Its performance was officially validated on 30 subjective databases. In the standardization process, it competed with six other proposals. Owing to its good performance and modular approach, parts of it have since been released in the official standard ITU-T Rec. P.1203.3, but this paper shows the full model as it was submitted.

In Section II we first give some background information on the P.NATS competition and related work that we considered in our research. Section III will focus on the model and its components; we explain how we trained and evaluated it in Section IV. Discussion and conclusions follow in Section V and Section VI, respectively.

II. Background and Related Work

A. ITU Standardization in Audiovisual QoE

ITU-T's SG12 has a long history in creating instrumental QoE models. Q14 as a subgroup specifically focuses on audiovisual streaming applications. In the scope of their work, several quality models have been developed in forms of competitions, including Recommendations P.1201 and P.1202 [1]. Those models are bitstream-based and allow for the prediction of streaming quality in the presence of packet loss. With the advent of HTTP streaming, Q14 had started work on P.NATS in the year 2014, just after releasing an update of ITU-T Rec. P.1201 Amd. 2, App. III, which enables the use of the P.120x models for progressive download applications.

The P.NATS group initially comprised eight proponents with both academic and industry background. The group collaboratively designed test plans and performance criteria for

the models that they would later evaluate. The main goals—as with the previous standards—included proper conduction of the underlying tests and careful validation of the models' performance, in order to release a well-tested standard.

B. QoE Models and Subjective Tests

Models for predicting QoE for HAS vary widely in their designs. Some use analytical methods, that is, closed-form equations in the shape of Score $= f(\vec{x})$, where $\vec{x}$ is a set of independent variables. [2], [3] are examples of such models, in which the sequence or its features (e.g., number of stalling events) are analyzed as a whole. Other models implement time series filtering [4] to predict the current quality at any given time in the sequence. Representations of the user's state of mind have also been used [5]. Finally, some models employ machine learning methods, for example Random Forests, as shown in [6]. In that work, the model is used to improve the rate adaptation of an HAS player.

Such QoE models make use of subjective tests, which serve several purposes: helping identifying the factors influencing QoE, choosing a modelling technique, training the developed model, and validating its performance. For example, Seufert *et al.* tested different methods of pooling scores over a longer time period [7]. The authors have later conducted studies on representation switching behavior and its effect on QoE [8], noting that more sophisticated pooling methods would have to take into account the time spent on a certain quality layer [9]. In those studies, however, sequences are typically rather short, leading to an unusually high amount of quality degradations for some test conditions. Robitza *et al.* [10] have addressed this issue by conducting tests that systematically study quality switching effects in longer video sequences, without repeating source contents. All of those tests however may not cover enough conditions to be able to prove whether (or how much) a certain factor influences user-perceived QoE. The use of databases from the P.NATS competition therefore enables us to more reliably look into the above effects.

III. Model Description

For our analytical QoE model we chose a modular approach with as few interdependencies as possible. This would allow us to extract certain components and combine them with other proponents' models at a later stage. Figure 1 shows the entire model. It receives lists of video and audio Mean Opinion Score (MOS) values, as well as an indication of all stalling/rebuffering events and their length.[1]

A. Preparation Module

In the *Preparation Module* we scale the input scores to a range of $[1, 5]$ (*Value scaling*) if necessary, and remove/duplicate scores such that there is one score per media second (*Temporal scaling*). In the following, N is the number of individual temporally scaled scores, that is, the length of the sequence in seconds, without buffering events.

[1]In practice, the MOS values can be calculated using existing models (ideally, the video module from ITU-T P.1203.1, but any estimation of quality can be used). Buffer events could be extraced from client-side logs (e.g., through player APIs) or estimated from network logs using buffer models.

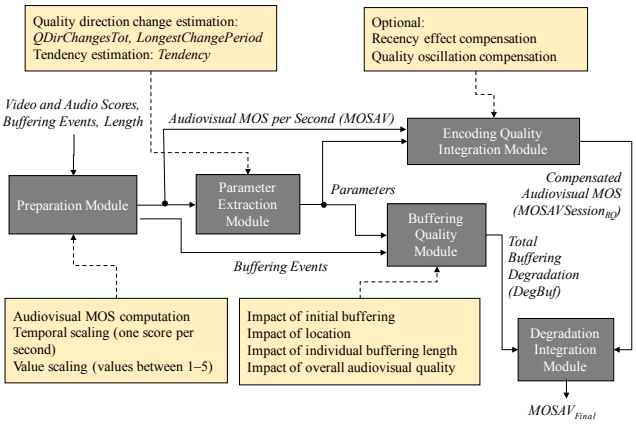

Fig. 1. Block diagram of the proposed model. Light boxes indicate steps inside modules.

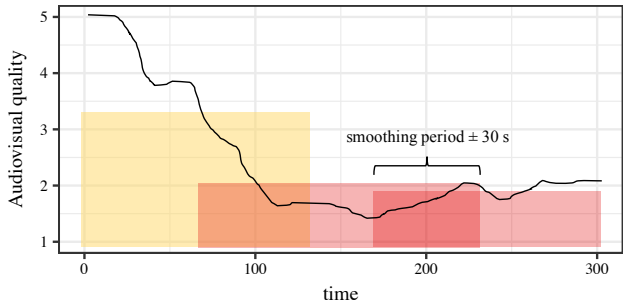

Fig. 2. Example of a sequence with "early drop" Tendency.

We follow the procedure according to ITU-T Rec. P.1201 Amd. 2, App. III to convert the scaled audio and video MOS scores a_i and v_i into "R-Scale" values[2] for $i \in [1, \ldots, N]$:

$$\begin{aligned} qCodA_i &= RfromMOS(a_i) \\ qCodV_i &= RfromMOS(v_i) \end{aligned} \tag{1}$$

The audiovisual quality scores are calculated as follows, taking into account video and audio quality in an additive and multiplicative fashion (P.1201 Amd. 2, App. III, Section 9.3):

$$\begin{aligned} qAV_i =& 100.867 - 0.3590 \cdot qCodA_i - 0.9210 \cdot qCodV_i \\ &+ 0.00135 \cdot qCodA_i \cdot qCodV_i \end{aligned} \tag{2}$$

We then calculate the per-second audiovisual MOS values $MOSAV_i$, which are used as input for our model:

$$MOSAV_i = MOSfromR\,(qAV_i) \tag{3}$$

B. Parameter Extraction Module

1) Tendency of Scores: One factor that influences the QoE is the overall *Tendency* of quality throughout the sequence. Does it start well but degrade after that? Is it continuously showing high quality? To gauge this, we developed the following algorithm: first, the sequence is temporally split into three equally sized parts, with 10% overlap ("smoothing period")

[2]The $RfromMOS$ and $MOSfromR$ functions are defined in ITU-T Rec. P.1203.1, Annex E. They transform scores from 1–5 to a range from 0–100 or vice-versa. The functions were originally used for the "E-Model" (ITU-T Rec. G.107).

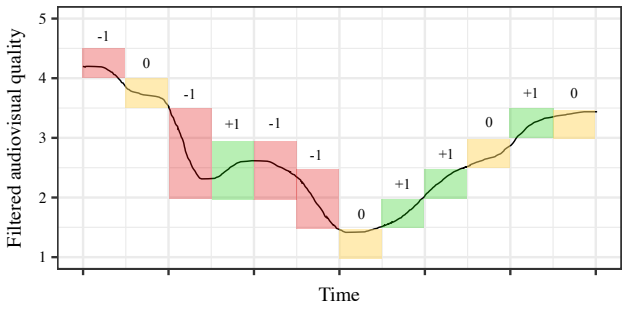

Fig. 3. Example of quality change determination. Thresholds are not to scale.

between the regions (see Figure 2). Then, the average of $MOSAV$ in each of those three parts is calculated and rounded to the nearest 0.5. We set the *Tendency* to *flatHigh* if all these means are ≥ 4.0, to *flatLow* if all are ≤ 2.5, and *earlyDrop* if the beginning is higher than the middle and the middle score equals to the end. Figure 2 shows a sequence with an "early drop". Note that other combinations of score averages are possible, leading to different tendency estimations. They were not used in this model, but may be further investigated and part of future model extensions.

2) Quality Direction Changes: In order to address quality variations over time as another significant impact factor, we count the number of quality direction changes (*QDirChangesTot*) in the entire sequence. To do this, we first calculate a moving average of order 5 on the $MOSAV_i$ scores. Note that in order to get a valid filter output for the start of the sequence, the first score in $MOSAV$ is copied four times and prepended to $MOSAV$. We then initialize an empty list *QDirChanges* and compare each score $MOSAV_i$ with $MOSAV_{i+3}$, incrementing i by 3 until we reach the end of the sequence. In other words, we only look at every third score and check the difference between it and the succeeding (third) score (i.e., look at the scores in 3-second-steps). If that difference is > 0.2, we record a 1 in *QDirChanges*. If the difference is < -0.2, we record a -1. If there is no difference (within the threshold), we record a 0. The step size of 3 was chosen to prevent minor quality variations from having a too strong impact. Note that we optimized the parameters chosen here based on a manual count of quality changes in the training sequences as we subjectively interpreted them.

The above process is shown in Figure 3, albeit with different thresholds for easier visualization. In this case, we would have recorded $[-1, 0, -1, +1, -1, \ldots]$ in *QDirChanges*. We can use these entries to count the number of quality direction changes: it is the sum of all consecutive 1s and -1s in that list. In the above case, the number of changes is equal to 6. The rationale for this number is—as we will see later—the more quality changes, the lower the QoE.

3) Longest Changing Period: Using the esimation of quality changes from the previous section (i.e., *QDirChanges*, the list of 1s, 0s, and -1s), we also want to find the longest period in which the quality is increasing or decreasing only. This is achieved with the following algorithm: iterating through *QDirChanges*, we count the number of steps as long as the quality is monotonically increasing or decreasing (i.e., going in one direction only), or staying the same. When the quality direction changes, the counting resets and starts from 0. The *LongestChangingPeriod* is then the overall maximum of those counted steps, multiplied by the step size. It is measured in seconds. If that period has a large value (e.g., around 60), it means that there were no quality oscillations in large portions of the sequence.

```
l = Vector
for index, direction in QDirChanges:
    if direction != 0:
        if (l is empty) or
        (direction of last element of l != direction):
            append [index, direction] to l
if l is not empty:
    prepend [0, 0] to l
    append [length of QDirChanges, 0] to l
    distances = Vector
    for each current and next element of l
        distance = next - current
        append distance to distances
    LongestChangingPeriod = (maximum of distances) * 3
else:
    LongestChangingPeriod = (length of QDirChanges) * 3
```

C. Encoding Quality Integration

In this module, the audiovisual scores $MOSAV_i$ are pooled into a final, single audiovisual quality score ($MOSAVSession$) that represents the QoE without taking into account rebuffering. First, we apply a simple averaging over the scores, meaning that:

$$MOSAVSession = \frac{\sum_i^N MOSAV_i}{N} \tag{4}$$

While training our model, we found that other advanced methods like increased weighing of lower scores or a Minkowski summation [7] did not (significantly) improve the prediction accuracy.

1) Recency Effect Compensation: The recency effect explains the fact that the last portion of a sequence has the strongest impact on the overall subjects' quality judgement [11]. It should be noted that in our considerations for modeling, this effect is assumed to be orthogonal to any other temporal effects, that is, the impact of quality variations over time, or the possibility of users "forgetting" what happened at the beginning of the sequence. However, in practice, there may be interaction effects.

To account for the recency effect, we included a component that weighs the last $MOSAV$ scores more than the rest. "Last" is defined as the so-called recency period. Its length L_r depends on the sequence length N, as we identified in the subjective test results:

$$L_r = \left\lceil 13 + 17/\left(1 + \exp\left(2 \cdot \left(5 - \tfrac{N}{60}\right)\right)\right)\right\rceil \tag{5}$$

We now calculate the average $MOSAV$ score within that period:

$$MOSAVRecency = \frac{\sum_{i=N-L_r}^{N} MOSAV_i}{N - L_r} \tag{6}$$

Note that $N - L_r$ marks the start index of the region after which we expect recency effects.

The recency compensation is not activated if (1) $MOSAVRecency \geq 3.1513$ or (2) if the *Tendency* is *flatLow*

or *earlyDrop*, because subjects would already give a bad score to such sequences and we wanted to prevent overcompensation. In this case, $MOSAVSession_R = MOSAVSession$.

Should the compensation be applied, we set a weight to every score within that recency period and calculate a weighted average of $MOSAV$ to produce a recency-compensated $MOSAVSession_R$, according to the following algorithm:

```
MOSAV_w = MOSAV
w = Vector of 0 with length N
oversum = 0
for index, score in MOSAV:
    if index >= N - L_r:
        k = i - (N - L_r)
        w[index] = exp(0.1016 * k)
        oversum = oversum + w[i]
for index, score in MOSAV:
    if index < N - L_r:
        w[index] = 1 - (oversum - L_r)/(N - L_r)
    MOSAV_w[index] = score * w[index] * 1/N
MOSAVSession_R = sum(MOSAV_w)
```

Here, the weight of every individual score exponentially increases as we reach the end of the sequence.

2) Quality Oscillation Compensation: In order to account for a large number of quality oscillations (i.e., frequent direction changes observed within $QDirChanges$), another component is introduced after the optional recency effect compensation.

If the longest changing period takes up a too large portion of the sequence ($\frac{LongestChangingPeriod}{N} \geq 0.25$) or if that period is longer than 30 seconds, no oscillation compensation is applied, meaning that $MOSAVSession_{RQ} = MOSAVSession_R$. With this check, we exclude long, monotonous changes that would otherwise be falsely identified as oscillations. Otherwise:

$$MOSAVSession_{RQ} = MOSAVSession_R - \min\big(\exp\left(0.3601 \cdot QDirChangesTot - 4.409\right), 1.5\big) \quad (7)$$

D. Buffering Quality Estimation

This module calculates the overall impact of degradation ($DegBuf$) due to initial loading ($DegInit$) and all stalling events ($DegStallTot$) within the sequence:

$$DegBuf = \max\big(\min\left(DegInit + DegStallTot, 4\right), 0\big) \quad (8)$$

1) Initial Buffering: For $DegInit$ we chose to re-use the equation from ITU-T P.1201 Amd. 2, App. III in our model:

$$DegInit = \begin{cases} \max\left(\min\left(0.29 \cdot \log\left(T_0 - 3.29\right), 4\right), 0\right), & T_0 > 4.29 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

2) Stalling Degradations: $DegStallTot$ is a sum of the degradation of each stalling event $DegStall_n$, with $n \in [1, \ldots, N_{stall}]$ for all stalling events:

$$DegStallTot = \max\Big(\min\big(\sum_{n=1}^{N_{stall}} DegStall_n, 4\big), 0\Big) \quad (10)$$

An individual stalling event's degradation is calculated as the following:

$$DegStall_n = w_{loc_n} \cdot w_q \cdot deg_n \quad (11)$$

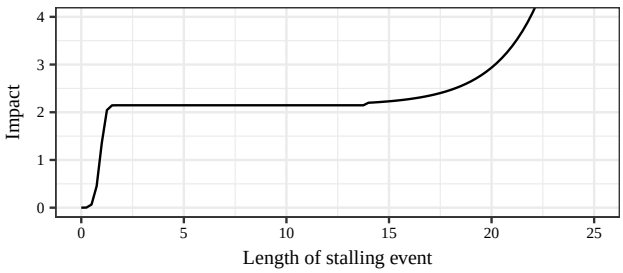

Fig. 4. Degradation impact of a single buffering event's length based on Equation 14.

Here, the weight w_{loc_n} is set based on the location loc_n of the stalling event in the sequence (in seconds). In the subjective tests we saw that this location could impact their perceived severeness. The further away from the end of a sequence, the lower their impact on quality:

$$w_{loc_n} = \max\big(-0.2566 \cdot \log\left(N - loc_n\right) + 0.5316, 0.01\big) \quad (12)$$

The weight w_q is based on the *Tendency* of the sequence, as obtained from the *Parameter Extraction Module*. If the sequence is of good quality throughout, subjects will rate stallings more negatively:

$$w_q = \begin{cases} 1.2073, & Tendency = flatHigh \\ 1, & otherwise \end{cases} \quad (13)$$

The actual degradation deg_n is calculated depending on the length of the stalling itself, len_n. It follows a combination of two psychometric functions:

$$deg_n = 2.1462 \cdot \begin{cases} 1 - exp\left(-x^5\right), & len_n < 10 \\ \min\big(\exp\left(0.45 \cdot len_n - 10\right) + 1, 4\big), & otherwise \end{cases} \quad (14)$$

Figure 4 shows a combined curve plot of the two functions of Equation 14. We can see how the function calculates a lower impact of shorter stalling events (i.e., stallings that would be imperceivable or not annoying enough), but exponentially increases after a given threshold. We empirically set thresholds and weights based on our observations.

E. Degradation Integration Module

In the *Degradation Integration Module*, we simply subtract the degradation due to initial loading and stalling ($DegBuf$) from the audiovisual MOS ($MOSAVSession$):

$$MOSAVFinal = MOSAVSession_{RQ} - DegBuf \quad (15)$$

This is the final MOS, ranging from 1–5, which corresponds to the user's QoE after having watched the sequence. Here, we assume that AV scores and buffering can be treated independently of each other, which can be considered a simplification.

IV. Training and Performance

A. Training Data

In the process of standardizing P.1203, the proponents created a total of 30 databases, split between 17 training

and 13 validation databases, comprising 1064 audiovisual sequences. The databases simulated typical HAS conditions with quality variations, initial buffering and rebuffering events, with sequence lengths between 1 and 5 minutes. 7 of the databases had the same sequences rated both on PC/TV screens and mobile screens. For an overview of the databases, see ITU-T Rec. P.1203, Section 8.

All P.NATS proponents had access to the same training databases, with which they could develop their model. Our model, as shown in this paper, was trained on only 9 of the 17 training databases—those that were shown on PC/TV screens—with a total of 334 sequences.[3] Since the databases are owned by the respective creators, we cannot show more details about the used content and conditions in this paper; we refer to [10] for a similar test design with respect to the rated conditions, contents, and the test protocol.

To instantiate our model, we needed audio and video quality scores for the sequences. We used the already standardized audio quality model from ITU-T Rec. P.1201 and our own video quality model, which is now standardized in ITU-T Rec. P.1203.1 Annex D (Mode 3). It operates on the bitstreams and extracts coding-related information on a per-frame basis; it then pools those data to output per-second video quality estimations, taking into account degradations due to lossy encoding, upscaling, and framerate reduction.

B. Training Procedure

In order to train a model such as the one presented in this paper, simple techniques like linear regression may not be applicable or suffice. For closed-form equations (e.g., the initial loading degradation from Equation 9) one would fit a parametric curve using data points obtained from subjective tests using non-linear least-square minimization techniques. However, in our case the model contains a relatively high number of parameters and algorithms. Because of this complexity, it is more likely that the minimization algorithm finds a local minimum only, yielding unstable parameter combinations. We therefore leverage the stochastic *Differential Evolution* algorithm to find a global optimum to the minimization problem, rather than using classical algorithms like *Levenberg-Marquardt* or *Nelder-Mead*.[4]

We developed a dedicated Python program that utilizes the `lmfit` package in order to minimize the error between training set data (i.e., subjective MOS) and the model output: Root Mean Square Error (RMSE) as specified in ITU-T Rec. P.1401, Section 7.5.1. It is the same measure that was used to validate the performance of submitted P.NATS models in terms of absolute prediction error. Our program was written in such a way that several model variants could be tested and evaluated against each other, with optional constraints defined on the parameter ranges. We empirically set these ranges before training runs, using observations we made on the subjective data. For example, we manually created exponential or logarithmic curves with certain parametrizations according to how we imagined the rough shape of the curve, then let those parameters vary by a predefined amount. This training was used for the coefficients in Equations 7, 9, and 12–14.

[3]Other databases were rated on mobile phones. ITU-T Rec. P.1203.1 specifies how the predicted MOS scores have to be adjusted in the case of mobile screens, but this adjustment is not part of this model and therefore beyond the scope of this paper.

[4]https://docs.scipy.org/doc/scipy-0.18.1/reference/tutorial/optimize.html

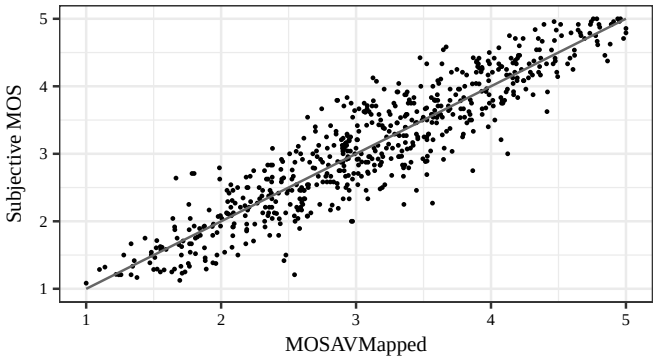

Fig. 5. Model prediction with linear fit per database against subjective MOS.

For parts of the model (i.e. the thresholds in Section III-B/C) an exhaustive (brute force) search for the optimum was computed. This was done before the actual training of the rest of the model's components using manually entered ground truth data. We obtained that data by counting our estimate of the quality changes ourselves. With this approach we could freeze some model parameters before the main training run.

C. Performance

We report the performance of the model that was submitted to the competition, meaning that its parameters were optimized on training databases only (i.e., we did not re-train our model after gaining knowledge of the validation databases). As stated before, the video and audio quality scores with which our model was instantiated correspond to ITU-T P.1203.1 (Mode 3) and P.1203.2, respectively.

For performance calculation, we apply the same procedure as adopted when standardizing P.1203. Similar procedures have been used in other standards, too, for example ITU-T P.863. In order to compensate for rating differences between subjective databases, we calculate a first-order linear fit between $MOSAVFinal$ and subjective MOS, per database, yielding:

$$MOSAVMapped_{i,k} = a_i * MOSAVFinal_{i,k} + b_i \quad (16)$$

where the coefficients a_i and b_i are calculated for every database i and every sequence k in that database. RMSE is then calculated between the fitted model output $MOSAVMapped$ and MOS. The overall average RMSE is 0.3859, ranging between 0.2028 and 0.5818 for all databases. The average RMSE for training and validation databases is 0.3432 and 0.4170, respectively. The overall Pearson correlation with subjective MOS is $r = 0.92$.

Figure 5 shows $MOSAVMapped$ and the subjective MOS values for all PC/TV training and validation databases. We can observe a good prediction accuracy over the entire MOS range, with only few PVSes being outliers. Note that agreements between proponents in P.NATS prevent us from comparing the performance of our model with other candidates.

V. Discussion

A. Future Directions

Our model uses a simple averaging of the per-second audiovisual quality scores to determine the final audiovisual quality, with optional compensation for the recency effect and quality variations. As suggested in [7], such a pooling method may suffice, and in our initial tests, none of the methods proposed there yielded significantly improved performance. Yet, other, more complex approaches like time series [4] or predicting a user state [5] could be worth investigating. More generally, the aim for future models would be not only predicting MOS, but also possible user reactions such as cancelling playback due to severe problems. Such reactions however are very context-dependent and proper test methods have to first be developed to study them.

B. Ecological Validity

The use of sequences with a length between 1 and 5 minutes constitutes a step in the direction of creating more ecologically valid tests, and we could observe that "forgetting effects" come into play 2–3 minutes after having seen a degradation. These effects could also be part of a pooling strategy, as seen in our buffering quality component. More generally, these findings prove that tests for HAS with sequence lengths of less then a minute will not show the full picture of QoE. It would be necessary to investigate even longer sequences or video sessions with multiple sequences, which corresponds to everyday usage of video on demand services. Related to this is an investigation of the impact of different viewing and usage contexts (e.g., home vs. mobile use, paid vs. free services) on the quality perception, which we hypothesize is primarily visible in the response to initial loading times and playback interruptions.

C. Sparsity of Test Designs

In the P.NATS process, all proponents created a comparably large number of databases, which treat different factors that impact HAS quality. Care was taken to design these databases systematically, for example by varying the number of quality switches and their depth, by creating symmetrical degradation conditions to investigate temporal effects (e.g., a quality drop at the beginning vs. the end of the sequence), or by applying the same degradation patterns to sequences of different lengths. However, even with over 1000 sequences, some MOS ratings could not be fully explained, meaning that we could not prove or disprove some theories of human perception, especially when related to combined effects of stalling and quality variations. Crowdsourcing studies may help in collecting even more data points for analysis, in terms of number of conditions and ratings. However, the design of such studies requires more careful preparation and screening of results. It remains to be checked whether the ratings from such tests are valid, and whether they can be combined with laboratory experiments, especially in the rigid context of standardization.

VI. Conclusion

In this paper we presented a QoE model developed for HAS, parts of which have become standardized in ITU-T P.1203.3. It integrates per-second audio and video quality scores with buffering events as measured at the client side, and predicts a final QoE value as experienced by the user. The model's input scores can be generated by any other existing quality model. For the prediction, we take into account temporal factors such as quality variations over time and the recency effect. We also consider the length and location of stalling events as well as the overall tendency of the scores. By using a modular approach, the model's individual components can be used in isolation (e.g., to describe certain features or quality impairments in a sequence) or retrained as parts of future models.

ITU-T SG12 Q14 is continuing research on quality prediction models for video streaming, together with the Video Quality Experts Group (VQEG). It also works on standardizing methods for evaluating the quality of entire video sessions, which will be the next step in creating reliable and valid QoE models for online streaming services.

Acknowledgment

This paper is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 643072. The authors would like to thank their colleagues at ITU-T SG12 Q14 for the collaborative effort in standardizing P.1203.

References

[1] A. Raake, J. Gustafsson, S. Argyropoulos, M.-N. Garcia, D. Lindegren, G. Heikkilä, M. Pettersson, P. List, and B. Feiten, "IP-based mobile and fixed network audiovisual media services," *Signal Processing Magazine, IEEE*, vol. 28, no. 6, pp. 68–79, 2011.

[2] T. Mäki, M. Varela, and D. Ammar, "A Layered Model for Quality Estimation of HTTP Video from QoS Measurements," in *Second Workshop on Quality of Multimedia Services, QUAMUS*, 2015.

[3] H. T. T. Tran, T. Vu, N. P. Ngoc, and T. C. Thang, "A novel quality model for HTTP adaptive streaming," in *2016 IEEE 6th International Conference on Communications and Electronics, IEEE ICCE 2016*, 2016, pp. 423–428.

[4] Chao Chen, Lark Kwon Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, "Modeling the Time-Varying Subjective Quality of HTTP Video Streams With Rate Adaptations," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2206–2221, May 2014.

[5] H. Yeganeh, R. Kordasiewicz, M. Gallant, D. Ghadiyaram, and A. C. Bovik, "Delivery Quality Score Model for Internet Video," in *ICIP*, 2014.

[6] Y. L. Chien, K. C. J. Lin, and M. S. Chen, "Machine learning based rate adaptation with elastic feature selection for HTTP-based streaming," in *IEEE International Conference on Multimedia and Expo*, 2015.

[7] M. Seufert, M. Slanina, S. Egger, and M. Kottkamp, "'To pool or not to pool': A comparison of temporal pooling methods for HTTP adaptive video streaming," in *QoMEX*, 2013, pp. 52–57.

[8] S. Egger, B. Gardlo, M. Seufert, and R. Schatz, "The Impact of Adaptation Strategies on Perceived Quality of HTTP Adaptive Streaming," in *VideoNext '14*, 2014, pp. 31–36.

[9] M. Seufert, T. Hossfeld, and C. Sieber, "Impact of intermediate layer on quality of experience of HTTP adaptive streaming," in *2015 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 256–260.

[10] W. Robitza, M.-N. Garcia, and A. Raake, "At Home in the Lab: Assessing Audiovisual Quality of HTTP-based Adaptive Streaming with an Immersive Test Paradigm," in *QoMEX*, 2015.

[11] R. Aldridge, J. Davidoff, M. Ghanbari, D. Hands, and D. Pearson, "Recency effect in the subjective assessment of digitally-coded television pictures," in *Fifth International Conference on Image Processing and its Applications*, 1995, pp. 336–339.