“To pool or not to pool”: a comparison of temporal pooling methods for HTTP adaptive video streaming

Michael Seufert, Martin Slanina, Sebastian Egger, Meik Kottkamp

Angaben zur Veröffentlichung / Publication details:

Seufert, Michael, Martin Slanina, Sebastian Egger, and Meik Kottkamp. 2013. “‘To pool or not to pool’: a comparison of temporal pooling methods for HTTP adaptive video streaming.” In *Fifth International Workshop on Quality of Multimedia Experience (QoMEX), 3-5 July 2013, Klagenfurt am Wörthersee, Austria*, edited by Christian Timmerer and Patrick Le Callet, 52–57. Piscataway, NJ: IEEE. https://doi.org/10.1109/qomex.2013.6603210.

"TO POOL OR NOT TO POOL": A COMPARISON OF TEMPORAL POOLING METHODS FOR HTTP ADAPTIVE VIDEO STREAMING

Michael Seufert [1,3]*, Martin Slanina* [2,3]*, Sebastian Egger* [3]*, Meik Kottkamp* [4*]

[1] University of Würzburg, Institute of Computer Science, Würzburg, Germany
E-Mail: seufert@informatik.uni-wuerzburg.de

[2] Brno University of Technology, Department of Radio Electronics, Brno, Czech Republic
E-Mail: slaninam@feec.vutbr.cz

[3] FTW Telecommunications Research Center Vienna, Vienna, Austria
E-Mail: egger@ftw.at

[4] Rohde & Schwarz GmbH & Co. KG, Munich, Germany
E-Mail: meik.kottkamp@rohde-schwarz.com

ABSTRACT

Current objective video quality metrics typically estimate video quality for short video sequences (10 to 15 sec) of constant quality. However, customers of video services usually watch longer sequences of videos which are more and more delivered via adaptive streaming methods such as HTTP adaptive streaming (HAS). A viewing session in such a setting contains several different video qualities over time. In order to express this in an overall score for the whole viewing session, several temporal pooling methods have been proposed in the related work. Within this paper, we set out to compare the performance of different temporal pooling methods for the prediction of Quality of Experience (QoE) for HTTP video streams with varying qualities. We perform this comparison based on ground truth rating data gathered in a crowdsourcing study in the context of the NGMN P-SERQU project. As input data for the models, we use objective video quality metrics such as PSNR, SSIM but also very basic inputs such as the bitrate of the clips only. Our results show that certain pooling methods perform clearly better than others. These results can help in identifying well performing temporal pooling methods in the context of HAS.

Index Terms— adaptive streaming, HTTP streaming, video QoE, temporal pooling, mobile video, variable video quality

1. INTRODUCTION

According to a recent study [1], already 51 percent of mobile traffic was video traffic by the end of 2012 and this share and its total volume are expected to rise in the next years. When transmitting videos over a wireless network, changing channel conditions are a big problem. Even though packet loss can be prevented by the use of TCP, different radio conditions and congestion lead to varying bandwidths. If the available bandwidth falls below the video bit rate, eventually stalling will occur which severly deteriorates the users' Quality of Experience (QoE) [2]. To avoid stalling in mobile environments, streamed videos either have to be encoded in a very low quality or adapt dynamically to the available bandwidth.

HTTP adaptive streaming (HAS) is a popular adaptation approach which reuses existing web technology and switches between different versions of the same content. On the server, the video is available in two or more different encodings and split into small chunks, each containing a few seconds of the video. The client requests the small chunks via HTTP and can choose between the different encodings depending on its current network condition. If channel conditions get worse, the rate determining algorithm in the HAS client adapts to the reduced bandwidth by selecting lower bit rate chunks. If the bandwidth increases, higher bit rate chunks can be requested. Thus, stalling is avoided, but the video properties, however, can change during playback in three dimensions. The video resolution can be decreased or increased (spatial dimension), the frame rate can be altered (temporal dimension), the encoding or compression can be changed (image quality dimension), or a combination of these three can be applied.

Video quality assessment tries to determine users' QoE from objective video quality metrics. Most of these video metrics are based on the image quality of the individual frames. By temporal pooling, these periodical measures of the video sequence can be aggregated over time to get one measure for the whole sequence. To predict the subjective

*The first and second author performed the work while they were affiliated with FTW Telecommunications Research Center Vienna.

quality of videos in classical (i.e. non-adaptive) video streaming, temporal pooling of objective frame metrics already has been used. In adaptive streaming scenarios, additional quality changes and sections with lower quality occur and impact users' QoE. However, they can be measured by objective frame metrics, too. Thus, in this paper we evaluate whether temporal pooling is suitable for HAS and which methods and which objective metrics are best to assess the subjective quality of adaptive video streams. Therefore, the paper is structured as follows: Section 2 presents related work and Section 3 describes the test setup. In Section 4 we present the compared pooling methods and objective metrics, and show the results. Section 5 concludes this paper.

2. RELATED WORK

2.1. HTTP Adaptive Streaming

HAS was first introduced by MOVE Networks in 2007, and since then has been implemented in proprietary streaming technologies such as Apple's HTTP Live Streaming[1], Adobe's HTTP Dynamic Streaming[2], or Microsoft's Silverlight Smooth Streaming[3]. Recently this approach also has been standardized by MPEG: Dynamic Adaptive Streaming over HTTP (DASH) [3]. A comparative approach in terms of achieved quality are the works [4, 5, 6] which inspect the performance of different HAS approaches. Another study [7] compares HAS performance to the performance achieved with the Scalable Video Coding (SVC) extension [8] of H.264/AVC which is a competing approach. In terms of varying video quality as a result of HAS studies, [9, 10] report quality scores for different video quality profiles. What is however missing in this related work is a thorough understanding of how certain different video qualities over time sum up in the a posteriori ratings of HAS sequences.

2.2. Temporal Pooling

Objective methodologies for video quality assessment typically output quality scores per frame. As analyzed video sequences consist of T frames, a certain (temporal) pooling metric is needed to combine these T scores to an overall score for the sequence. An early comparison of simple pooling methods is given in [11]. They found that pooling methods which more strongly weigh the most degraded and/or most recent parts of a sequence perform best. Moreover, for metrics with scores in a narrow dynamic range (such as SSIM), they found that the pooling method does not influence the overall quality score at all. However, they used only short video sequences with little motion, thus, their results might not be applicable to typical HAS content.

More sophisticated pooling methods are presented in [12, 13, 14, 15, 16]. A different pooling approach using spatial pooling is described in [17]. However, all mentioned temporal pooling methods are mainly targeted towards video sequences of 10 to 15 sec and miss out pooling approaches for longer video sequences as typically viewed in HTTP adaptive streaming sessions. In order to close this gap, we want to 1) *analyse if current pooling methods can be paramaterised to work for longer video sequences* and 2) to answer the question *which of these pooling methods performs best?*

3. TEST SETUP

3.1. Video Quality Profiles

As source sequences we used two 100 sec clips from the open source video project Sintel[4]. Each of these clips consisted of twenty 5 sec video chunks which were available in six different bitrate settings as described in Table 1. The target bitrate was achieved by encoding the chunks with different quantization parameters, thus, while streaming, only image quality was adapted. The content of both clips was animation with a high degree of details and several scenes with high motion intensity. We have chosen to take into account only one content class for this analysis as we wanted to basically understand the relationship between different quality levels, and obtain a summative overall rating for the whole clip. As 20 video chunks with six video quality levels already span a huge multi-dimensional space of 6^{20} combinations, we wanted to stay focused with this one content class.

In the NGMN P-SERQU project [18], these videos were streamed via HAS technology (Apple HTTP Live Streaming) through a representative wireless network. Typical mobile LTE conditions were emulated by adjusting network factors such as fading, round trip time, interference/noise, jitter, and competing traffic. As some of the resulting video profiles (i.e. videos including quality adaptations) were similar, pruning of the test set allowed to insert artificial profiles, i.e., profiles that did not show up while using the Apple HLS client in the test network, but could occur with other HAS technology. The resulting video experiences (16 realistic and 22 artificial profiles for *Clip1*, 40 realistic and 12 artificial profiles for *Clip2*) were stored and used in the crowdsourcing study. They contained 0-20 quality adaptations, but no stalling.

3.2. Crowdsourcing Setup

Within the QoE research community, crowdsourcing attracts increasing attention as a novel methodology for conducting subjective user studies. In essence, the subjective test is outsourced to a large anonymous crowd of subjects, who re-

[1] http://developer.apple.com/resources/http-streaming/

[2] http://www.adobe.com/de/products/hds-dynamic-streaming.html

[3] http://www.microsoft.com/silverlight/smoothstreaming/

[4] http://www.sintel.org

Quality Level	1	2	3	4	5	6
Target Bitrate [kbit/s]	128	210	350	545	876	1410

Table 1. Quality Levels and Target Bitrate settings

motely complete the test at their own computers. Especially micro job platforms like Amazon's Mechanical Turk[5] or Microworkers[6] can be used to recruit test users as described in [19] and [2]. For subjective video quality tests, specialized frameworks exist like [20] and [21]. To this end, participants launch a web-based application in their browser and click through the subjective test. The main advantage of crowdsourcing are low costs and especially the speed at which tasks and test campaigns are completed. However, since the users are conducting the test remotely without direct supervision, reliability of test participants is not guaranteed, especially if payment is used as an incentive. This is the major challenge for crowdsourcing QoE assessment as described in [2].

In order to overcome these reliability issues we have chosen only voluntary crowd users throughout partnering institutions of the NGMN P-SERQU project. To ensure that the users experience the desired test conditions, the video profiles (i.e. videos including quality adaptations) were prepared offline and these videos were offered as Podcasts. In order to participate, the test user had to download a Podcast containing five different video profiles on her iPhone or iPad. During the download phase, a personal data questionnaire including consistency questions was completed by the participant. After that, the user had to click a button for starting the test, which appeared upon download completion. Then the user sequentially viewed all video clips. After the playback of each video, the user was asked to rate her current personal satisfaction with the video quality on a 5-point ACR scale.

During the five month test phase, 494 users participated. For our evaluation, we selected those 297 users who completed the whole test on an iPad.

4. RESULTS

4.1. Per frame quality metrics

The basis of the temporal pooling analysis is formed by the per-frame values of objective video metrics OM. In this work, two full reference metrics have been used, namely the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM). Both metrics have been calculated for the luma component of the two considered video clips. Moreover we used trivial objective metrics which depend on properties of the chunk to which a frame belonged. Thus, we assigned to each frame the quality level (i.e. values from 1 to 6, cf. Table 1) or the bitrate of the respective chunk. Additionally for each quality level we obtained the mean of the MOS of constant quality level profiles (i.e. profiles without adaptations) and assigned this metric to the frames.

[5] http://www.mturk.com/mturk/welcome

[6] http://www.microworkers.com/

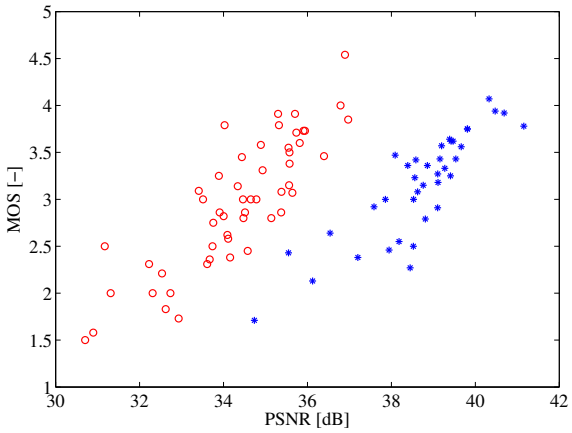
4.2. Preprocessing

The simplest and quite natural temporal pooling approach is to calculate the mean value of the objective metric output over all frames of the video clip. Then, one can calculate the correlation between these mean values and the MOS scores gathered in the crowdsourcing experiment. After a quick look at the mean values of PSNR for different profiles evaluated for the two clips (see Fig. 1(a)), it was obvious that the mean PSNR values for all the profiles of the two clips are in a different range: For *Clip1*, the mean PSNR over all quality profiles was in the range [34.7, 41.2] dB while for *Clip2*, the mean values were in the interval [30.7, 37.0] dB. This is caused by the different spatio-temporal characteristics of the two clips. As a result, the mean objective metric values, as well as outputs of other temporal pooling algorithms, cannot be directly mapped to MOS scores for the two sequences at once.

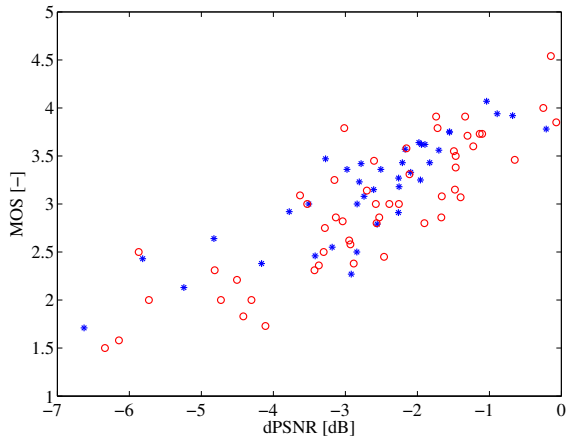
Assuming that the encoder is optimized for perceptual quality and that the perceptual quality at level 6 (Tab. 1) can be considered as Excellent on the MOS scale, the difference in objective metric values can be compensated by subtracting the mean values of the level 6 sequence from all per-frame PSNR values for all sequences. The input to the temporal pooling algorithms can then be expressed as *objective metric value with respect to mean value of highest available quality in the sequence* as shown in Fig. 1(b).

4.3. Temporal pooling

In the experiment, 13 different pooling methods were used and applied to the objective metrics OM. Six methods were described in [11]: For histogram pooling (Histogram) the k-th percentile of the cummulative histogram values is used. Low values of k express the influence of the lowest quality frames on viewers. For Minkowski summation (Minkowski, $[\frac{1}{T} \cdot \sum_{t=1}^{T} OM^p(t)]^{1/p}$) and exponentially-weighted Minkowski summation (Exp-Minkowski, $[\frac{1}{T} \cdot \sum_{t=1}^{T} \exp(\frac{t-T}{\tau}) \cdot OM^p(t)]^{1/p}$) high values of p emphasize the influence of highest quality frames. Exp-Minkowski additionally accounts for recency effects by the exponential weighing factor with parameter τ. Mean pooling (Mean, $\frac{1}{T} \cdot \sum_{t=1}^{T} OM(t)$) and last frames mean pooling (MeanLastFrames, $\frac{1}{F} \cdot \sum_{t=T-F}^{T} OM(t)$) simply compute the mean of all or respectively the most recent F frames' objective metrics. Local minimum of mean values of N successive frames (LocalMinimum, $\min[\frac{1}{N} \sum_{i=1}^{N} OM(t+i)]$) emphazises the influence of the poorest quality section on the overall score. A related approach but not described in [11] simply computes the mean of the p percent of overall frames with lowest quality (Percentile).



(a) Mean PSNR value.



(b) PSNR difference from best sequence.

Fig. 1. Scatter plot diagram of MOS versus mean objective metric values for all profiles under consideration and *Clip1* (blue stars), *Clip2* (red circles).

In [12] the hysteresis effect is presented. To account for that, for any frame the minimum of quality scores over the last τ seconds is combined with an element that indicates the current quality. The mean of these values is the overall score (Hysteresis, see detailled description in [12]). In [13] two parametric functions are described which, similar to Minkowsi pooling, transition continuously from mean ($p = 0$) to max pooling ($p \to \infty$): SoftMax ($\sum_{t=1}^{T} \frac{\exp(p \cdot OM(t))}{\sum_{u=1}^{T} \exp(p \cdot OM(u))} OM(t)$) and LogExp ($\frac{1}{p} \log \frac{1}{T} \sum_{t=1}^{T} \exp(p \cdot OM(t))$). In [14] objective metrics are clustered by k-means algorithm into two clusters, containing frames of lower and higher quality respectively. After reducing the impact of the less important higher quality frames by multiplying a weight, the scores are combined (KMeans, see detailed description in [14]). A similar approach is sequence level pooling (SequenceLevel, see detailed description in [15]) which divides the frames according to a percentage parameter instead of clustering. In [16] the score is computed from the mean of the objective metrics and the differences between successive frames. This score emphasizes quality deteriorations and can decrease down to a saturation threshold (VQA, based on detailed description in [16] but slightly modified).

4.4. Performance Comparison

As described above most of the metrics require one or more input parameters to account for recency effects and/or to emphasize low quality which both have a higher impact on subjective quality. But no a priori limitation of the parameters is suitable because it is the aim of the experiment to examine the usability of each temporal pooling algorithm for adaptive video streaming. Thus, all the parametric temporal pooling algorithms are tested for a number of parameter values. This assumption leads to an optimization problem: we need to find the optimal parameter values for a given temporal pooling algorithm which maximize the correlation of the temporal pooling output with the subjective scores.

As the dataset is limited to two video clips only, the optimization needs to be cross-validated in order to make sure that the pooling algorithm is performing well independently of the training set (used to adjust the parameters) and the evaluation set (used to check the performance of the trained algorithm) selection. For cross-validation, the leave one out algorithm was used, which is known to be the most exhaustive among all cross-validation algorithms. Its principle is such that during the training, one sample is removed from the whole set of sequences and the training is done on the rest. Then, the output for the single left out sample is evaluated. Using this approach as many times as there are samples in the whole dataset, one gets the same number of cross-validated outputs as there are samples in total. A good temporal pooling algorithm should exhibit high correlation of the cross-validated values with the MOS scores for the corresponding profiles.

In the training phase, we have used two different optimization criteria - the Pearson Linear Correlation Coefficient (PLCC) and Spearman Rank Order Correlation Coefficient (SROCC) which were also used in the evaluation to assess the performance of the different temporal pooling algorithms.

The results of the training with cross-validation are given in Tab. 2 for temporal pooling of per-frame PSNR and SSIM values. The best performing methods are Percentile, Mean, VQA, SequenceLevel, and Hysteresis which perform well for both PSNR and SSIM. With Minkowski and SoftMax pooling the special case exists that they can be identical to Mean pooling for certain parametrizations and thus could reach better results which can be seen in the Mean row. However, to keep the specific characteristics of the methods visible we excluded this special case in the table. Also with LocalMinimum the special case was excluded that the minimum of the mean of

all successive frames was the mean of the last frames. Thus, also LocalMinimum could reach better results which can be seen in the MeanLastFrames rows.

Pooling algorithm	PSNR		SSIM	
	PLCC	SROCC	PLCC	SROCC
Percentile	0.854	0.835	0.854	0.864
Mean	0.851	0.837	0.870	0.866
VQA	0.846	0.833	0.820	0.767
SequenceLevel	0.834	0.777	0.790	0.815
Hysteresis	0.826	0.802	0.867	0.859
Histogram	0.782	0.781	-0.004	-0.150
LogExp	0.571	0.538	0.817	0.812
ExpMinkowski	0.565	0.549	0.572	0.513
LocalMinimum	0.555	0.447	0.506	0.511
MeanLastFrames	0.485	0.477	0.437	0.445
KMeans	0.353	0.334	0.192	0.263
SoftMax	0.259	0.297	0.686	0.648
Minkowski	-0.058	0.368	0.815	0.842

Table 2. Correlation between the pooled objective values (PSNR, SSIM) and subjective MOS after training and cross validation (sorted by PLCC of PSNR pooling).

Finally, Tab. 3 presents the result achieved when quality levels or MOS values of constant quality levels (i.e. MOS of constant level profiles) were used as inputs to the temporal pooling. For the temporal pooling in this case, all we need is the scheme of quality level switching along all the profiles and optionally the six representative MOS values. The results are comparable to those achieved from temporal pooling of per-frame objective scores, in some cases even better. Only KMeans pooling failed to classify the input of quality levels probably because of the very small number of distinct values. Apart from that, also the worst PSNR and SSIM pooling methods perform quite well with quality level and constant quality level MOS pooling. Both more trivial metrics also outperform the pooling of chunk bitrates which only reaches correlations up to 0.79 for the best methods. It is worth mentioning that simple mean pooling of quality levels is one of the best performing methods, although it is the most general approach which includes almost no information about the underlying content. As the output of pooling of constant quality level MOS score is on the same scale as the subjective ratings, the scatter plot diagram representing the SequenceLevel pooling, reaching the highest correlation in this scenario, is presented in Fig. 2. It illustrates that a good performance can be achieved by temporal pooling for all 90 quality profiles without any extreme outliers.

5. CONCLUSION

Throughout this paper we have investigated the performance of different temporal pooling methods on their prediction performance for longer video sequences with durations in the magnitude of minutes. In order to gain maximal performance of the compared metrics, we have in a first step optimized their parameters for longer video sequences and then compared their performance for different inputs such as PSNR, SSIM, quality levels and quality level related MOS.

Pooling algorithm	Quality Level		Constant QL MOS	
	PLCC	SROCC	PLCC	SROCC
SequenceLevel	0.861	0.851	0.883	0.857
VQA	0.867	0.860	0.870	0.882
Mean	0.856	0.837	0.864	0.842
Percentile	0.824	0.806	0.864	0.863
Minkowski	0.820	0.800	0.843	0.818
MeanLastFrames	0.829	0.805	0.842	0.811
ExpMinkowski	0.822	0.802	0.833	0.804
Histogram	0.785	0.789	0.802	0.738
LogExp	0.746	0.711	0.797	0.751
LocalMinimum	0.831	0.803	0.793	0.761
Hysteresis	0.766	0.726	0.780	0.753
KMeans	-	-	0.739	0.748
SoftMax	0.656	0.598	0.724	0.662

Table 3. Correlation between the pooled quality levels and pooled constant quality level MOS scores and subjective MOS after training and cross validation (sorted by PLCC of Constant QL MOS pooling).

Our results show that the performance of the certain pooling methods reaches good levels taking into account that they were intentionally not developed for longer video sequences as used within this analysis. Surprisingly, we also found that pure mean (of objective metrics) performs on par with the best performing pooling methods. Regarding the input used for the pooling methods, we found that even very basic information such as the different quality levels of the video chunks in a sequence yield to prediction performance comparable to the performance gathered by using far more complex inputs such as objective metrics like PSNR and SSIM. This means that an ultra low complexity metric as the mean of quality levels of a HAS video sequence delivers already very decent prediction performance. However, for the initial assignment of quality levels to frames, PSNR/SSIM analysis qualifies due to similar correlation results.

To pool or not to pool? To answer this question, we need to be aware of the setup in which we have obtained the objective scores per frame or per chunk. For sequences of length in the order of minutes, we have shown that none of the sophisticated pooling algorithms performs significantly better than a simple mean of values. Such finding advocates the use of mean value as a single objective quality representative in such scenario. Using the mean of objective scores, one can obtain a good overall quality estimate for a several minutes long sequence.

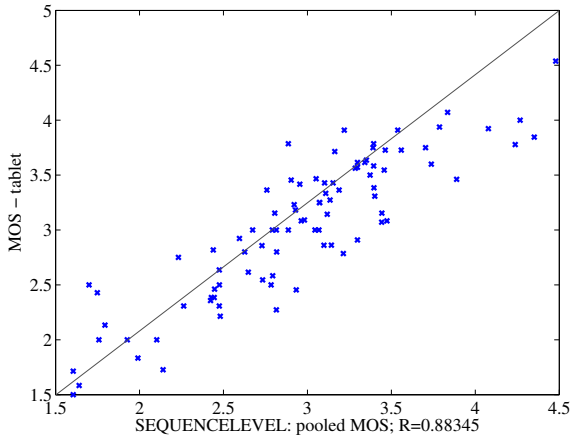


Fig. 2. Scatter plot diagram of MOS versus model output based on pooling of MOS values for all profiles under consideration.

6. ACKNOWLEDGMENTS

The information provided is based on work carried out in the on-going NGMN Project SERQU; the work in the NGMN Project SERQU is not finalized yet. Any information provided is preliminary and subject to change in a final approved NGMN document on SERQU. A part of this work has been performed within the projects U-0 and ACE 2.0 at the Telecommunications Research Center Vienna (FTW) and has been funded by the Austrian Government and the City of Vienna within the competence center program COMET. In addition this work was supported by the project CZ.1.07/2.3.00/30.0005 of Brno University of Technology and also by the SIX project; the registration number CZ.1.05/2.1.00/03.0072, the operational program Research and Development for Innovation.

7. REFERENCES

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017," Tech. Rep., Cisco, 2013.

[2] T. Hoßfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, and P. Tran-Gia, "Quantification of YouTube QoE via Crowdsourcing," *IEEE International Workshop on Multimedia Quality of Experience*, 2011.

[3] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, 2011.

[4] C. Müller, S. Lederer, and C. Timmerer, "An evaluation of dynamic adaptive streaming over HTTP in vehicular environments," *4th Workshop on Mobile Video*, 2012.

[5] J. Yao, S. S. Kanhere, I. Hossain, and M. Hassan, "Empirical evaluation of HTTP adaptive streaming under vehicular mobility," *10th international IFIP TC 6 Conference on Networking*, 2011.

[6] B. Lewcio, B. Belmudez, A. Mehmood, M. Waltermann, and S. Moller, "Video quality in next generation mobile networks – Perception of time-varying transmission," *International Workshop Technical Committee on Communications Quality and Reliability*, 2011.

[7] H. Kalva, V. Adzic, and B. Furht, "Comparing MPEG AVC and SVC for adaptive HTTP streaming," *IEEE International Conference on Consumer Electronics*, 2012.

[8] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, 2007.

[9] N. Cranley, P. Perry, and L. Murphy, "User perception of adapting video quality," *International Journal on Human-Computer Studies*, 2006.

[10] J. D. McCarthy, M. A. Sasse, and D. Miras, "Sharp or smooth?: comparing the effects of quantization vs. frame rate for streamed video," *SIGCHI Conference on Human Factors in Computing Systems*, 2004.

[11] S. Rimac-Drlje, M. Vranjes, and D. Zagar, "Influence of temporal pooling method on the objective video quality evaluation," *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2009.

[12] K. Seshadrinathan and A. C. Bovik, "Temporal hysteresis model of time varying subjective video quality," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011.

[13] Y. Boureau, J. Ponce, and Y. Lecun, "A Theoretical Analysis of Feature Pooling in Visual Recognition," *27th International Conference on Machine Learning*, 2010.

[14] J. Park, K. Seshadrinathan, S. Lee, and A. C. Bovik, "Video Quality Pooling Adaptive to Perceptual Distortion Severity," *IEEE Transactions on Image Processing*, 2013.

[15] K. Lee, J. Park, S. Lee, and A. C. Bovik, "Temporal pooling of video quality estimates using perceptual motion models," *17th IEEE International Conference on Image Processing*, 2010.

[16] A. Ninassi, O. Le Meur, P. Le Callet, and D. Barba, "Considering Temporal Variations of Spatial Visual Distortions in Video Quality Assessment," *IEEE Journal of Selected Topics in Signal Processing*, 2009.

[17] J. You, J. Korhonen, and A. Perkis, "Spatial and temporal pooling of image quality metrics for perceptual video quality assessment on packet loss streams," *IEEE International Conference on Acoustics Speech and Signal Processing*, 2010.

[18] NGMN, "Service Quality Definition and Measurement (P-SERQU)," 2012, `http://www.ngmn.org/workprogramme/service-quality.html`.

[19] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with Mechanical Turk," *SIGCHI Conference on Human Factors in Computing Systems*, 2008.

[20] K. Chen, C.-J. Chang, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "Quadrant of euphoria: a crowdsourcing platform for QoE assessment," *IEEE Network*, 2010.

[21] C. Keimel, J. Habigt, C. Horch, and K. Diepold, "QualityCrowd – A framework for crowd-based quality evaluation," *Picture Coding Symposium*, 2012.