QoE-aware optimization for video delivery and storage

Alisa Devlic*, Pavan Kamaraju†, Pietro Lungaro*, Zary Segall*, and Konrad Tollmar*
*Mobile Service Lab, Royal Institute of Technology (KTH), Kista, Sweden
†Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, USA
Email: devlic@kth.se, pavan4@umbc.edu, pietro@kth.se, segall@kth.se, konrad@kth.se

***Abstract*—The explosive growth of Over-the-top (OTT) online video strains capacity of operators' networks, which severely threatens video quality perceived by end users. Since video is very bandwidth consuming, its distribution costs are becoming too high to scale with network investments that are required to support the increasing bandwidth demand. Content providers and operators are searching for solutions to reduce this video traffic load, without degrading their customers' perceived Quality of Experience (QoE). This paper proposes a method that can programmatically optimize video content for desired QoE according to perceptual video quality and device display properties, while achieving bandwidth and storage savings for content providers, operators, and end users. The preliminary results obtained with Samsung Galaxy S3 phone show that up to 60% savings can be achieved by optimizing movies without compromising the perceptible video quality, and up to 70% for perceptible, but not annoying video quality difference. Tailoring video optimization to individual user perception can provide seamless QoE delivery across all users, with a low overhead (i.e., 10%) required to achieve this goal. Finally, two applications of video optimization: QoE-aware delivery and storage, are proposed and examined.**

I. Introduction

The proliferation of Internet-connected devices that can deliver video services provided over Internet rather than over a managed service provider's network and a lot of video content available online caused explosion of Over-the-top (OTT) video traffic that is congesting operators' networks and degrading users' experience. Video accounts for more than 60% of Internet traffic and is expected to generate over 80% of global consumer traffic by 2018 [4]. The amount of video content that crosses global IP networks increases at rapid pace: over 38.2 billion free videos were watched during second quarter of 2014, which is 43% increase from the same quarter last year [3]. 60% of those videos were viewed on a smarthone and is expected that the traffic from wireless and mobile devices will exceed the wired device traffic in the near future [4] [9].

Bandwidth in cellular networks is a limited and expensive resource, which has to be shared among large number of users. Since video is very bandwidth consuming, its distribution costs became too high to scale with network investments that are required to support the increasing bandwidth demand. With arrival of 4K and 8K ultra high definition video quality, having 4 and 16 times more pixels than full HD, delivering videos to devices such as smart TV and tablets might create bottlenecks even in other types of networks (WiFi and wired) [10].

According to Conviva report from 2013, 60% of video streams in 2012 have experienced one of the following quality degradations: buffering interruptions, slow video startup, or low picture quality [6]. This investigation also showed that users are becoming impatient and intolerant about poor video performance that they experience and quickly switch to another source if the video quality they experience is not satisfactory. Failing to address these challenges and improve the viewers' experience, the video content providers risk to loose their subscribers, which will affect their and subsequently the operators' revenues. Additionally, despite the memory becoming larger, cheaper, and popularity of cloud services, there is a limit on storing videos in HD quality on OTT devices (especially on smartphones and tablets), which is easy to fill given the accessibility to fast Internet connectivity and increasing amount of video content. Therefore, there is a need for novel solutions that can reduce video traffic load and video storage requirements, without perceptible quality degradation.

The perceived video quality varies through video content, despite being encoded for the target bitrate, due to the nature of video content that can change from one scene to another. By removing this quality fluctation, we can potentially reduce video size without perceptual quality degradation. Moreover, delivering the video in the resolution higher than the device supports does not increase perceptual video quality[1] further than of the maximum supported resolution, while it increases the cost of transmitting and storing these extra bits. To address these challenges, this paper proposes a novel video optimization method that can optimize video for viewing on a mobile device based on *perceptual quality of short video segments* that are encoded in the maximum device supported and downscaled resolutions, thus reducing the video size without compromising perceived QoE[2] for a viewer. Preliminary results obtained with a smartphone show that this method can save bandwidth and reduce storage space without perceived quality degradation.

Videos can be optimized for a large user population or an individual user's perception, delivering average or personalized perceptual video quality, hence with potentially additional bits required in the latter case. These advantages inspired QoE-aware video storage and delivery applications that enable seamless QoE in viewing video, while reducing video storage and bandwidth requirements. Furthermore, combining perceptual video quality with data rate channel properties enables QoE-aware adaptive video streaming, which can maximize perceived QoE for the given video and access channel conditions and deliver the same or better quality than Dynamic Adaptive Streaming over HTTP (DASH) [11] with fewer number of bits.

The remainder of paper is organized as follows: Section 2 gives an overview of related work, followed by description of video characteristics and techniques for measuring perceptual video quality in Section 3. Section 4 explains methodology and

978-1-4799-7492-4/15/$31.00

[1] Represents the user's opinion about the video quality after seeing the video.
[2] The perceived QoE determines viewer's satisfaction with the video quality.

experiments required by video optimization, while bandwidth savings from optimizing movies are provided in Section 5. Section 6 illustrates potential gains and costs of tailoring video optimization to individual user's perception. Section 7 describes QoE-aware video storage and delivery applications. Finally, section 8 briefly discusses obtained results, followed by conclusion and plans for future work.

II. Related work

QoE-aware video optimization has been studied for some time context of video streaming, by balancing between video quality and available network resources.

Z. Li et al. [16] implemented video optimization as a dynamic programming algorithm which is repeatedly applied over a finite window size to deal with bandwidth variations. However, their solution trades the minimum buffer size for minimizing video quality variations, using large buffers to compensate for stability of segments' video quality. Instead, our proposed QoE-aware streaming uses initial buffer of 2 seconds, enabling video playout to start immediately after this time. Additionally, their optimization is myopic in terms of optimal solution computed by dynamic programming, having no insight into the overall perceptual video quality during video stream optimization, which might result in larger number of bits than neccessary spent to achieve the final video quality.

QDASH integrated an intermediate level into the bitrate switching process for gradual change of quality levels instead of abruptly switching down to the target quality level [18]. Similarly to [16], their adaptation strategy is myopic - it does not consider the overall perceptual video quality for optimizing video, before it is streamed to the user's device.

D. Miros and G. Knight implemented a video optimization method that manipulates a video stream bitrate by balancing between the content quality score, transmission rate, and sender's & receiver's buffer sizes [17]. The method consists of a neural network that predicts the video quality of a real time video based on 6 consecutive frames at a time, and a fuzzy rate-quality controller that considers these quality predictions when manipulating the video stream bitrate, in order to provide a smooth streaming quality. The quality predictions obtained from neural network are highly correlated with VQM scores, and being based on 6 frames, this method can be used in our work for obtaining objective video quality scores online.

Rückert et al. used VQM scores to optimize the Scalable Video Coding (SVC) [23] video quality according to user perception, reducing up to 60% bandwidth required to stream video in P2P-based video on demand system [22]. Their proposed optimization sets the layer that *maximizes* perceptual video quality as the target optimization layer. The advantage of SVC over AVC is that it can switch on-the-fly between different video qualities, resolutions, and frame rates during streaming session according to network conditions. With separate caching of content partitioned in different layers and large number of users requesting the same content, SVC improves caching performance and cache-hit ratio of the system.

We decided to implement our method using AVC, because there are no efficient SVC decoders available on mobile devices and DASH-enabled SVC has not been standardized. An experimental H.264/SVC multi-core decoder has been implemented and evaluated on Android 4.0, however showing that decoding videos at higher resolutions on a smartphone decreases the frame rate [15], which degrades a user's QoE. Our method requires frame rate to be fixed (at 24fps), while varying frame resolution, which cannot be achieved with their solution. It can potentially be applied to the encoded layers of SVC video, by merging parts from predicted layers only (into a single composed layer) based on perceptual video quality.

Frame rate reduction combined with adaptive quantization has been used in [5] to reduce the bandwidth associated to video streaming. Their optimization approach quantifies the amount of motion in MPEG video stream and drops the frames of video scene in case of low motion, while for high motion scene reduces the quality of frames by changing quantization level. Subjective tests showed that this method can improve perceptual video quality by 30% at different bandwidth fluctuation rates and motion characteristics.

Note that video optimization by resolution reduction, performed according to perceptual quality on a mobile device, has not been used to optimize a video, streaming it to the user's device over DASH to the best of our knowledge. Hence, it has been shown in [24] that a small decrease of frame resolution is better perceived than the frame rate reduction by the same percentage and results in higher bitrate reduction! Additionally, neither of the found related works considered *proactively optimizing a video stream* (before the streaming starts) *for the optimal perceptual video quality* that can be achieved in the given video and access channel conditions in order to reduce the video stream size. The proposed video optimization method is not restricted to streaming, optimized video can also be provided in a file for download or prefetching.

III. Video characteristics and user perception

A. Spatial and temporal video information

A video represents a sequence of frames captured over time, usually accompanied by an audio track. Video frames have, therefore, a spatial and a temporal component. The spatial information represents the amount of spatial detail of a picture. It comprises the appearance of objects in the picture, resolution, smoothness, complexity of textures, transitions in intensity and color hue (known as contrast). The temporal information indicates the amount of temporal changes in video sequence. It represents the measure of motion of objects in a video or movement of background including scene changes.

Spatial and temporal information are perceived by humans by distinguishing, for example, action clips from slow moving clips, or scenes with complex textures and higher contrast from scenes with large monotone surfaces and smooth transitions.

Temporal information can be related to *frame rate*, while spatial information maps to *frame resolution*. Higher frame rate leads to smoother movement in a video, while higher resolution increases video sharpness. These parameters are also used as encoding parameters. The product of frame rate, resolution, and color depth is **bitrate**:

$$\begin{aligned} bitrate[bit/s] &= frame\_size * frame\_rate \\ &= resolution * color\_depth * frame\_rate \end{aligned} \tag{1}$$

Note that by reducing video bitrate, the amount of details in video decreases, reducing also the video file size.

B. Perceptual video quality

As depicted in (1), video bitrate can be reduced by decreasing frame size and/or frame rate. Content type determines how much frame rate or frame resolution can be reduced without degrading the perceptual quality of video. For example, a higher motion video tolerates lower frame quality on the expense of higher frame rate, in order to enable smooth motion of players. While low motion video, such as news, tolerates a lower frame rate, but requires a higher frame resolution to compensate for accurate representation of motion [26]. Viewers also have more time to pay attention to details, due to fewer changes between the successive video frames.

The frame resolution determines the number of pixels in each video frame. If pixels are spread over a large area on the screen, the perceived sharpness decreases. Alternatively, when pixels are squeezed into a small area, the image gets smaller and the perceived sharpness increases. Therefore, if video is played on the full screen, a **device display size** and **its pixel density** play a large role in determining the *minimum required frame resolution that can provide the best perceived video quality*. Increasing the resolution over this value *will not increase* the perceived video quality further, while it will increase video size and bandwidth required to deliver this video to the end user.

Since each video is accompanied by an audio track, audio quality also affects the perceived video quality. However, earlier work has shown that visual attributes contribute more to quality perception than aural properties [25]. Since the focus of this work is on video quality optimization, modifying audio properties to reduce bitrate is out of scope of this work.

Besides resolution, content type, and device screen size, perceptual video quality depends on other factors too, such as: viewing distance of the observer from the video, brightness, contrast, sharpness, naturalness, and color [28]. Any of these factors can be considered to identify parts of the video where video bitrate can be reduced, without the Human Visual System perceiving the quality distortions.

The video optimization proposed in this paper **reduces the resolution of video frames** whose spatial and temporal quality attributes remain the same (or slightly degrade) compared to frames in the original video quality, thus reducing the video file size and achieving potential bandwidth savings. T.Zinner and his colleagues showed that the quality of video sequences with reduced resolution is better perceived by users than the quality of video sequences at lower frame rate [30]. Additionally, reduction of frame rate saves less bandwidth and degrades more video experience than reduction of spatial resolution. These results also suggest that **video optimization** should be achieved **by reducing resolution** rather than the frame rate.

The perceptual video quality is commonly captured using Mean Opinion Score (MOS)[3] [12], a five point scale used to rate absolute and relative quality of multimedia content. Absolute video quality refers to evaluating video quality without a reference video, while relative video quality represents quality degradation of impaired video when compared to the reference video. Video sequences are shown to a panel of users, whose opinion is recorded and averaged into MOS. This procedure is referred to as subjective evaluation video quality test.

[3]Similarly to perceptual video quality, represents an average user's opinion about the video quality

Subjective video quality tests are the most accurate method to measure the perceptual video quality, since human perception represents the highest authority in evaluation of video quality. However, these tests are expensive in terms of time and human effort that need to be spent for their preparation and execution. In order to reduce this effort, objective video quality metrics have been developed, i.e., mathematical models that try to approximate results of subjective video quality assessment, based on criteria that can be evaluated by computer program. The performance of these metrics is evaluated by correlating objective video quality scores with MOS grades.

C. Video Quality Metrics

Video Quality Metrics (VQM) was the first video quality assessment method whose correlation with MOS grades exceeded 0.9, which resulted in standardization of this metrics by ANSI in July 2003 and later inclusion in ITU-T specifications [20]. Compared to other metrics which include PSNR, SSIM and other variants of these algorithms, VQM was the only model that exceeded 0.9 threshold, given by Pearson correlation coefficient. Therefore, it was chosen to compute the perceptual video quality of optimized videos in our work.

VQM represents an automated video quality measurement system that is based on linear regression of technology independent parameters closely approximating how people perceive video quality. These parameters are extracted from spatio-temporal (S-T) regions of the video sequence. VQM takes the source clip and the processed clip as input and computes the score using a series of steps. The first step includes division of video into S-T regions and application of perceptual filters to compute the perceptual video quality. In the second step, features are extracted for S-T regions, while in the third step VQM score is calculated by thresholding values obtained from the extracted features. VQM scores have scale from 0 to 1, with 0 being closest to the original video source.

The "general model" of National Telecommunications & Information Administration (NTIA) software [19] was used to compute VQM score, since this model is optimized to achieve maximum correlation between the objective and subjective video quality scores. Since the feature extraction from the original and processed clip requires high computation power, our model preprocesses the video to obtain the VQM score, *before* initiating video optimization.

VQM score is computed for a 4 to 15 seconds long video clip, by temporally and spatially collapsing its behavior, and *estimating the worst performance* that can be achieved by processing this video. We extended the idea of computing VQM score of a video clip to computing VQM score of each S-T region in the clip, in order to identify perceptual video quality of all regions. Having VQM scores of all S-T regions in a video enables video optimization, as described in Section IV. Furthermore, VQM scores are mapped to MOS grades to enable programmatic control of perceptual video quality.

IV. Video optimization

This section explains how video optimization works. Firstly, it requires as an input a video encoded in different resolutions (in our case 720p, 480p, 360p, and 240p), VQM scores for each of the video resolutions, and a target video quality threshold. The latter is expressed as a MOS grade or a VQM score, which can be obtained from linear regression

of VQM scores to MOS values, depicted in Figure 1 (more details about this model can be read in Section IV-A).

All videos in this paper are extracted from ten most popular movies in 2012. The movies were first ripped from Blu-ray discs to MKV format without loss of video quality. Next, they were encoded using VP8 video codec and FFmpeg tool in webm format to four resolutions (720p, 480p, 360p, 240p) following the YouTube recommendations for video encodings [1], without audio, and using Group of Pictures (GOP) of 6 frames. The movies were split into 15 seconds long clips and ran through VQM measurements to obtain VQM scores for each video resolution. The size of S-T region was 6 frames, since it was shown in [29] that this region size achieves maximum correlation with subjective ratings. VQM scores were obtained for each S-T region and the entire video clip.

The video optimization works by identifying the appropriate resolution for each 6 frames of video, comparing this chunk's VQM score in each downscaled resolution (starting with the lowest, 240p) with the VQM threshold, until finding the score that is lower than the given threshold. If the target score is not found, the original resolution of the video chunk (720p) is kept.[4] The consecutive video chunks with the same identified resolution are referred to as an optimized segment. This procedure is displayed in the pseudocode IV.1, with the video encoded in 720p (original video sequence) and in the following downscaled resolutions: 480p, 360p, and 240p.

Algorithm IV.1: VIDEOOPTIMIZATION($vqmScores, quality$)

$vqmThreshold \leftarrow regression(quality)$
$vqmScores_{480} \leftarrow vqmScores(1)$
$vqmScores_{360} \leftarrow vqmScores(2)$
$vqmScores_{240} \leftarrow vqmScores(3)$
for $i \leftarrow 1$ **to** $length(vqmScores_{480})$
do
- **if** $vqmScores_{240} < vqmThreshold$
  - **then** $optimizedRes(i) = 240$;
  - **else if** $vqmScores_{360} < vqmThreshold$
  - **then** $optimizedRes(i) = 360$;
  - **else if** $vqmScores_{480} < vqmThreshold$
  - **then** $optimizedRes(i) = 480$;
  - **else** $optimizedRes(i) = 720$;

return $(optimizedRes)$

After identifying all the optimized video segments, they are cut from the respective video resolutions, resized to the resolution of the original video, and merged into a webm file using FFmpeg tool [2] that can be played in the VLC media player on the smartphone. In case of video streaming, the segment cutting, resizing, and merging are omitted. Instead, a manifest file is created specifying list of optimized video segments in different resolutions and their byte ranges. In both cases DASH was used to split a video into 6 frames-based segments that can be taken from different resolutions and joined into a single video stream.

We performed subjective video quality tests on fixed resolution and optimized video clips, correlating the obtained results with VQM measurements in order to obtain VQM to MOS mapping. This mapping was used to derive VQM thresholds (representing the maximum VQM score for the perceptual video quality specified by MOS grade) or to compute MOS grade of an optimized video given its VQM score.

A. Experiments

Subjective video quality assessment tests were implemented using a Double Stimulus Impairment Scale (DSIS) method[5], evaluating the *perceptual video quality difference* between a reference and an impaired clip that were presented to users in the same test sequence. The original sequence was a video clip encoded in 1280x720 (720p) resolution, which was displayed on Samsung Galaxy S3 with high definition display resolution (720p) and maximum brightness set on the phone. The impaired sequence was the same clip with downscaled segments in resolution, also displayed in full screen.

An Android app was developed to perform the experiments, displaying video clips in VLC media player and enabling users to rate a perceptual quality difference between a pair of video clips. Each user was asked in the app, before displaying video clips, to enter information about their age, gender, and if they wear glasses or contact lenses. By associating a unique identity code to each user, all data was stored anonymously in the database and used for processing. Any incomplete or duplicate user's votes were discarded in the offline processing.

The training of subjects was conducted before starting a test, using an Android application - by showing a test pair of clips and voting procedure to the subjects, enabling them to familiarize with the test. Oral instructions were provided, explaining the purpose of the test, task, and the rating scale.

The experiment was performed with 32 people: researchers and students. 87.5% of test participants were men and the age of subjects ranged from 20 to 33 years old, with a median of 24 years. 50% of the participants had contact lenses or glasses.

Six different video clips were shown to the users, each in four resolutions: 720p, 480p, 360p, 240p and three perceptual video qualities indicated by VQM thresholds 0.11, 0.21, and 0.31 that were selected in order to result in MOS grades between 3 and 5 (which were acceptable to most of users). The order of resolutions and sequences of different video clips was randomized, in order to minimize the potential learning effect of video quality impairments. After viewing a pair of videos, users were asked to assess the quality difference using the following questions and selecting one of the answers:

1) Did you see a difference in quality between two clips?
   - a) Yes, first video had higher quality
   - b) Yes, second video had higher quality
   - c) No, they look the same
2) How did you perceive a difference in quality?
   - 4 Perceptible but not annoying
   - 3 Slightly annoying
   - 2 Annoying
   - 1 Very annoying

The answers obtained from the users were mapped to the impaired MOS scale and averaged to get a MOS grade of

[4] Note that by original resolution of the video the maximum supported resolution of the mobile device is assumed, which is equal to the device display size.

[5] A small deviation from the original DSIS method was made such that a pair of clips was presented to the user once (instead of twice as specified by P.910 recommendation), however allowing the user to view any of the two clips again before the voting. Additionally, for each video clip in one of the pair combinations (randomly chosen) the original sequence was inserted instead of the impaired one, in order to better evaluate the perceptual video quality difference.

each impaired video clip. MOS=5 was assigned to answer 1c) in which case the voting for the given video sequence was completed, while answers 1a) and 1b) indicated that the voting continued to the second question, where a user could assign a grade from 1 to 4 to perceptual video quality difference.

Figure 1 shows results of subjective tests after being correlated to the respective VQM scores. Linear regression was also recommended by Video Quality Experts Group as a fitting model of VQM scores to MOS grades [20], with a difference that they did not use optimized video clips in their tests. In our tests we considered clips in fixed resolutions **and** the optimized videos (which are displayed in red and blue color, respectively).

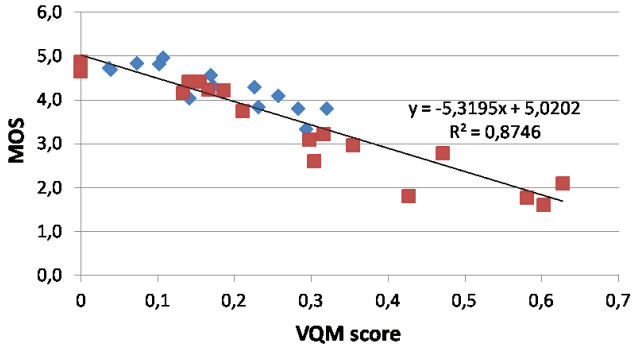

Fig. 1: Linear regression of VQM scores and MOS grades obtained from user experiments

The minimum and maximum δ related to computation of MOS grades from user votes were 0.1 and 0.4, respectively. Note that each user could have their own linear regression curve constructed based on his/her votes, enabling tailoring video optimization to their own perception, as it is illustrated in Section VI. However, by aggregating votes from all users in the regression curve we can programmatically control the perceptual video quality that can be applied to *any* user. This programmatic control is achieved by mapping MOS grades to VQM thresholds and using these thresholds to select appropriate resolutions for the optimized video segments.

Figure 2 illustrates MOS grades and file sizes of three video clips in different qualities that were used in subjective video quality assessment tests. It can be observed that MOS grades follow logarithmic function of the clips' file sizes, including clips in fixed resolutions and optimized video clips. This enables **estimation of bandwidth savings of a given video clip that is optimized for the particular MOS grade**.

The logarithmic function of MOS grades and video file sizes indicates that MOS increases at a slower pace as the video size becomes larger and larger. This means that in order **to notice an increase in perceptual video quality (equal to 1 MOS unit), one needs to spend increasingly more bytes when moving from lower to higher video qualities**. This relationship between quantity and intensity is known in literature as Weber-Fechner's law [14] and has previously been observed in relations of QoE with other QoS parameters [21] [8]. Figure 2 illustrates this behavior with a number of bytes required to reach the highest quality (MOS=5) from the quality below (MOS=4) being much higher than to reach the latter quality (MOS=4) from the quality below it (MOS=3). A large bandwidth gap between the existing video resolutions

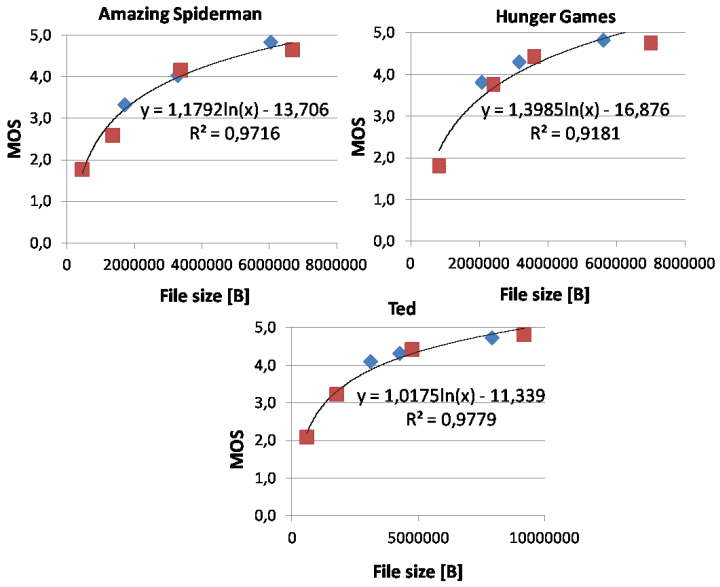

Fig. 2: Logarithmic relation of MOS and video clip's file size

(especially the largest ones) motivates the need for more operational points that can be used by content providers to reduce their costs and provide bandwidth & storage savings to their users, without compromising their perceived QoE.

To demonstrate potential bandwidth gains of additional operational points optimized for different perceptual video qualities, we applied the optimization method to entire movies, comparing them to the same files in reference resolution.

V. Optimization Method Applied to Entire Movies

This section proposes a method that can compute bandwidth savings from optimizing a long video (i.e., movie) to specific video quality. As with any video optimization, first step is to encode the movie in four resolutions, dividing it into 15 seconds segments. Next, the size of each optimized video segment for desired perceptual video quality is determined. Finally, sizes of all optimized segments are summed up and compared to the size of movie in the reference resolution.

Note that in this study, bandwidth savings are computed on optimized videos offline, *independently* of the network bandwidth or transport protocol used to deliver these videos to the user. Section VII-B presents a method that integrates video optimization with adaptive video streaming to deliver optimized videos in real time to the user over a varying data rate channel. The obtained file size reductions can also reduce video storage on a user's device, as discussed in Section VII-A.

A. Linear interpolation method

During optimization of several video clips we observed linear fit of VQM scores and associated file sizes on segments between consecutive fixed resolutions (illustrated with blue dashed lines in Figure 3). This observation inspired us to use linear interpolation to quickly compute an optimized video clip size, from the size of video in fixed resolutions and their VQM scores, without the need to perform video optimization.

One of the main challenges of this method was if interpolated points can actually be achieved using video optimization.

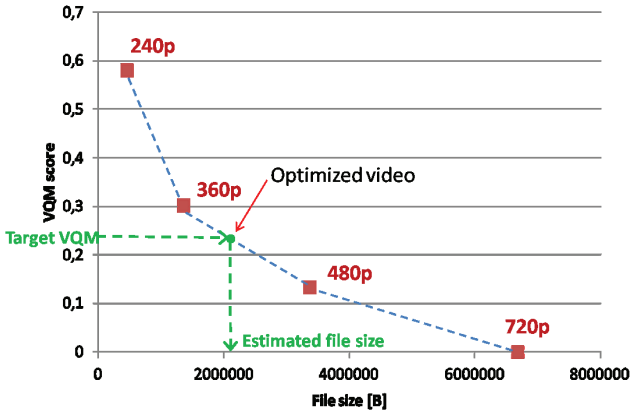

Fig. 3: Linear interpolation method to compute an optimized video's file size

To verify this, we interpolated Amazing Spiderman video size for various VQM scores, resulting in blue points in Figure 4.

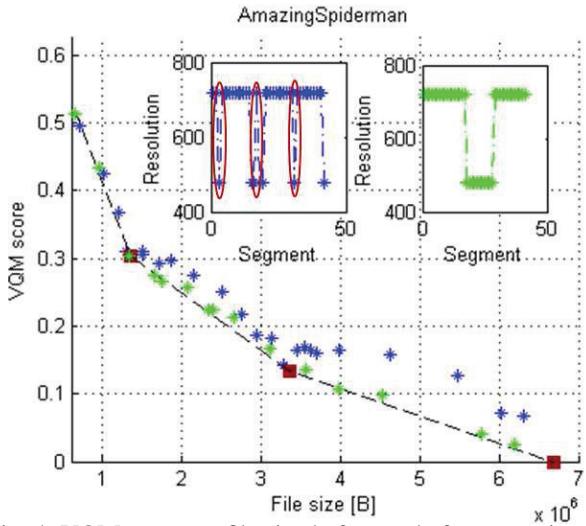

Fig. 4: VQM score vs. file size before and after removing short peaks

However, as it can be observed, these points ended up following not a linear, but more an elliptical curve. After inspecting resolutions and durations of different segments in optimized video clips, we noticed a pattern of short peaks in different resolutions that were 0.25-0.5 seconds long, appearing in optimized video clips and causing large deviation from the expected value in the linear fit. In comparison with this, the points that lied (close to) or overlapped with the expected points from the linear fit were composed of segments that were at least 1 second long. This made us suspect that these short peaks might have degraded the resulting VQM score. However, users did not perceive this quality degradation in tests, as shown in Figure 2, probably due to human eye inertia.

To verify this assumption, we removed all short peaks in different resolutions, requiring that a segment needs to be at least 0.75 seconds long. After this modification in the video optimization algorithm, the interpolated clip's points correlated well with estimated points from the linear fit, as displayed with green points in Figure 4. The prediction error was up to 8.6%.

According to Algorithm IV.1 the optimized video segments' quality will always be equal to or better than the desired video quality, causing an optimized video's VQM score to be lower than VQM threshold (in most cases). Our goal is to reach (or come as close as possible to) desired threshold with this score. This can be achieved by iterating video optimization, adjusting VQM threshold to the score obtained in previous iteration.

Figure 5 shows VQM scores obtained from applying the iterative video optimization algorithm to Amazing Spiderman video clip with various VQM thresholds. The monotonicity of this curve indicates that a **desired VQM score (or the closest achievable score) can be reached in finite number of steps by adjusting the VQM threshold value**. Additionally, results in Figure 4 and Figure 5 prove that using the linear interpolation method we can **compute achievable bandwidth savings for any video optimized for particular video quality**.

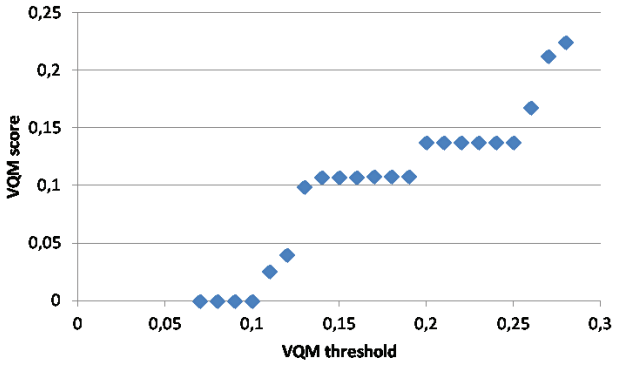

Fig. 5: Monotonic optimization curve of VQM threshold vs. VQM score

B. Determining reference movie resolution

By linearly interpolating each movie segment from the VQM scores and file sizes of this segment in two consecutive resolutions that are closest to the desired VQM threshold, we can compute the size of entire movie that is optimized for the given quality by summing up the interpolated segments sizes.

In order to compute potential bandwidth savings of the optimized movie, the reference movie resolution needs to be determined. The *reference movie resolution* is found as the minimum resolution of all the movie segments, whose 5% worst quality is higher than the perceptual video quality of the optimized movie. In VQM terminology this means that the 95^{th} percentile of all movie segments' VQM scores in this resolution needs to be lower than the target VQM threshold.

C. Performance measures and results

Bytes saved from optimizing a movie for the particular MOS grade are calculated as a difference between the size of the movie in reference resolution and the optimized movie size. **Bandwidth savings** is then computed as the ratio of the bytes saved by optimizing the movie for particular quality and the size of the movie in reference resolution.

The proposed video optimization requires videos to be encoded and cut on 6 frames bases at the frame rate of 24 fps and resized to the screen size resolution, before merging

the chunks and storing the optimized video into the file. If one would like to avoid this extra effort of video processing, especially during optimization of long videos such as movies, one could select each video clip's resolution (which is 15 seconds long) based on VQM scores of this clip in different resolutions, then merge these clips into the movie stream. We refer to the former way of optimizing the video stream on 6 frame bases as **micro optimization**, and to the latter that is based on 15 seconds as **macro optimization**.

Figure 6 shows the Amazing Spiderman movie size estimated for different MOS grades using micro and macro optimization in blue and green color, respectively. The movie sizes in reference resolutions are depicted in red color.

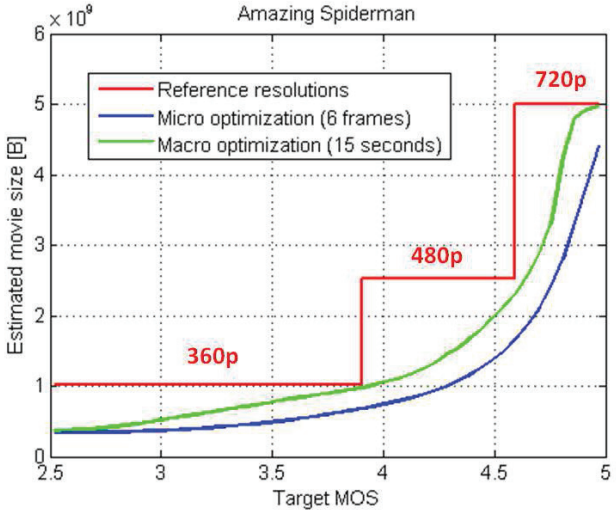

Fig. 6: Estimated size of Amazing Spiderman movie optimized for different MOS grades

Bandwidth and bytes saved by optimizing Amazing Spiderman movie for different MOS grades are illustrated in Figure 7, for micro and macro optimization. Observe that there are three reference movie resolutions for the selected target video qualities that are used to determine bandwidth savings of the optimized movie, illustrated by three line segments starting from MOS=5: 720p, 480p, and 360p.

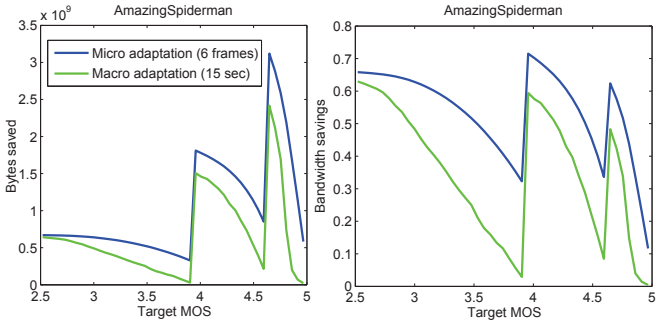

Fig. 7: Bandwidth and bytes saved from optimizing Amazing Spiderman movie for different perceptual video qualities

Figure 7 shows that even with macro optimization (i.e., by introducing VQM scores into the existing video quality-blind content delivery, without cutting and reencoding 6 frame segments) up to 50% bandwidth savings can be achieved by optimizing a movie to MOS=4.5 and up to 60% by optimizing a movie to MOS=4. On top of these bandwidth savings, by performing micro optimization additional 10% savings can be achieved (up to 60% and 70%, respectively). These micro adaptation savings correspond to 3GB and 1.8GB file size, which are larger than the size of the movie in 480p and 360p resolution, respectively. This can be explained by the fact that video optimization might compose segments using all four resolutions, which can in aggregate yield larger savings than it can be achieved by downscaling resolution of the movie file.

Bandwidth savings and bytes saved from optimizing other movies are given in Table I. Reference movie resolutions for all the movies are 480p and 360p for MOS=4.5 and MOS=4, respectively, for all movies except Avengers, which has the reference movie resolution of 480p for both MOS=4.5 and MOS=4. The results show that up to 57% bandwith savings can be achieved with macro optimization, and up to additional 12% (in total 69%) bandwidth savings can be reached with micro optimization. This illustrates the impact of introducing video quality score into the video optimization, which has so far been only QoS-oriented (i.e., videos have been encoded for the target bitrate and adapted during delivery to fit the currently available bandwidth, without concerning their perceptual video quality). Using video optimization to store optimized videos and deliver them to the end user according to their perception and preferences, can increase the users' perceived QoE, while saving network bandwidth and storage space on users' devices.

VI. Tailoring Video Optimization to Individual User's Perception

Video optimization that was described in previous section was based on linear regression of VQM scores to MOS grades that were computed *from all users' votes*. However, using MOS values to determine perceptual video quality levels might result in video quality being perceived *lower than expected* by more quality sensitive users, or *higher than requested* by quality insensitive people. Therefore, in this Section we construct the individual users' linear regression curves, evaluating potential advantages of tailoring video optimization to individual user's perception in terms of quality gains as well as potential costs in terms of extra bits that are required to achieve this goal.

A. Performance evaluation results

The linear regression curves constructed from individual users' votes and the corresponding VQM scores are depicted in Figure 8 in red color, while the linear regression curve built from all users votes' and VQM scores is shown in green. It can be observed that for a given VQM score, an individual grade (SOS) can deviate up to ± 2 points from the MOS value.

To evaluate the potential advantages of personalized video optimization, we define the accomplished perceptual video quality gain of each individual user (*ExtraMOS*) as:

$$ExtraMOS = fixedMOS - SOS, \qquad (2)$$

where *fixedMOS* represents the target grade for which the video needs to be optimized and *SOS* refers to the video quality optimized according to individual user's perception.

In order to determine how personalized video optimization affects bandwidth savings and assigned network resources, we

TABLE I: Bandwidth and bytes savings from optimizing movies with macro and micro optimization for MOS=4.5 and MOS=4

Movie	Macro, MOS=4.5		Macro, MOS=4		Micro, MOS=4.5		Micro, MOS=4	
	Bw savings	Bytes saved	Bw savings	Bytes saved	Bw savings	Bytes saved	Bw savings	Bytes saved
Avengers	35%	1.9GB	27%	0.7GB	54%	2.9GB	49%	1.3GB
Batman	51%	3.1GB	44%	1.4GB	66%	4GB	62%	1.9GB
Iron Man	51%	2.4GB	45%	1GB	65%	3GB	62%	1.4GB
Prometheus	57%	2.6GB	54%	1.2GB	69%	3.1GB	68%	1.6GB
Skyfall	45%	2.3GB	31%	0.8GB	60%	3.1GB	55%	1.4GB
Ted	30%	1.2GB	19%	0.4GB	51%	2GB	46%	0.9GB
Expendables	56%	2.1GB	57%	1GB	69%	2.6GB	69%	1.3GB
Hunger Games	46%	2.4GB	46%	1.2GB	62%	3.2GB	63%	1.6GB
Twilight Saga 2	54%	2.3GB	54%	1.2GB	69%	2.9GB	68%	1.5GB

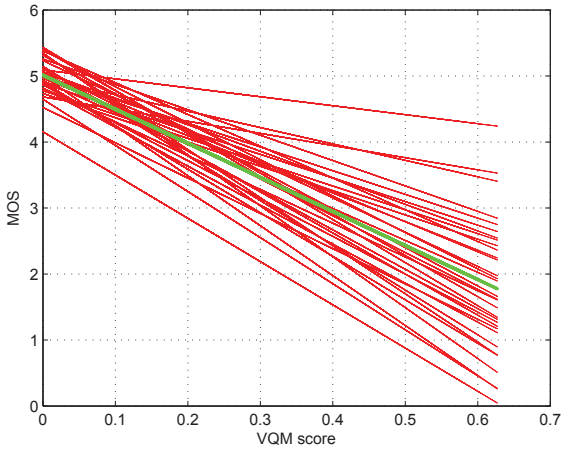

Fig. 8: Individual and aggregated users' linear regression curves

computed the number of additional bytes (*ExtraBytes*) that are required to satisfy all users with the delivered video quality:

$$ExtraBytes = \sum_{i=1}^{N} (userVideoSize_i - populationVideoSize), \quad (3)$$

where *userVideoSize* and *populationVideoSize* represent the size of the video optimized for individual and aggregated users' perception, respectively.

Figure 9 illustrates *ExtraMOS* computed for each user to achieve target grade 4 when optimizing Amazing Spiderman movie for invididual user's perception. Values above zero show extra video quality that needs to be provided to more quality demanding users, while negative values show how much below the average value video quality can be degraded while still satisfying the less quality sensitive users.

Figure 10 shows *ExtraBytes* that are required to optimize the Amazing Spiderman movie for target quality 4, in order to satisfy each user's QoE. Values above zero represent the additional bytes that are required to satisfy the perceptual quality of more quality sensitive users, while negative values represent bytes that can be saved from downgrading the quality of less demanding quality users.

The total of additional 2.97GB are needed to provide the desired video quality (i.e., MOS=4) to all users. However, our analysis shows that User 16, illustrated with the highest peak in Figure 9 and Figure 10, is an extreme user who gave arbitrary votes to clips optimized for different video qualities. If we exclude this user from the calculation of *ExtraMOS* (by setting

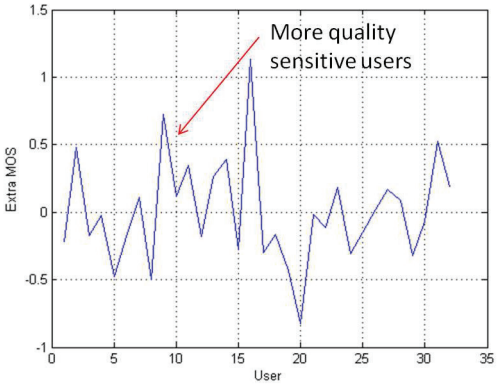

Fig. 9: Gains in video quality from tailoring optimization of Amazing Spiderman movie to individual user's perception, targeting MOS 4

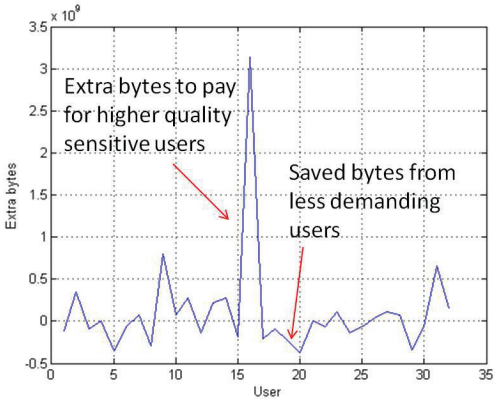

Fig. 10: Extra bytes required to optimize Amazing Spiderman movie for all users, targeting MOS 4

the requirement MOS$\pm$2SOS), 140MB would be saved with the personalized video optimization. Note that the amount of additional/saved bytes depends on desired video quality (higher qualities typically require higher resolutions and greater movie sizes) and balance of quality sensitive and less demanding users (lying to the right and to the left of the aggregated regression curve) in the target video quality.

VII. Applications of Video Optimization

A. QoE-aware video storage

QoE-aware video storage can be implemented as a multimedia file system on a user's device that downloads and stores optimized videos according to the user perception, preferences, video and device characteristics. The file system passes the

mapping function of VQM scores to SOS grades along with desired video quality (referred to as a user's QoE model) to the web server or the cloud system, that in turn optimizes the video, making it available for download to the user's device.

This file system is envisaged to be connected to the video quality assessment application on a user's device that can derive QoE model from subjective tests with the user.

B. QoE-aware video delivery

Video optimization can be seen as preprocessing of video content that usually occurs before the video is transmitted to the user's device. However, it assumes that optimized video files are (progressively) downloaded to the user's device. Depending on the bandwidth that is available to the user's device, if it is lower than the encoded video bitrate, it might take a while until enough of video is downloaded for it to start playing. Therefore, such a video delivery does not allow real time video viewing, unlike the video streaming.

Video streaming manages the video delivery and playback through video requests, where video is played as it is streamed to the device, without actually being stored at the user's device. However, streaming the video that is preprocessed in advance might cause interruptions in playback, if bandwidth cannot support the bitrate of a video segment that is being downloaded. To avoid playback interruptions and enable smooth playout, dynamic adaptive streaming over HTTP (DASH) downloads video segments in the highest quality that is below the achievable data rate. Hence, for this to work, video needs to be available in multiple bitrates to which the streaming can switch in case of lower bandwidth.

QoE-aware video delivery is envisaged to work on top of DASH, enabling a video stream to be optimized for target video quality in the same step as video is transmitted to the user's device. The key to maximize a user's QoE during video streaming is in *selecting the optimal video quality for which the video should be optimized* and streamed as such, without the need to often switch to lower bitrates due to insufficient bandwidth or lack of segments in the buffer. The optimal target video quality is, therefore, selected as the highest video quality that the available bandwidth can support without ending up in buffer underrun. However, if it happens that the entire video segment cannot be downloaded in the target quality until the end of next second, the proposed QoE-aware delivery switches to DASH streaming at lower qualities[6] in order to prevent playback interruptions. Additionally, instead of maximizing the individual segment's bitrate each second to use the entire throughput (as it is currently done by DASH), the difference between available bandwidth and the optimized segment's bitrate is used to *prefetch future seconds of the optimized video stream*, in order to prepare for potentially bad channel conditions and *prevent frequent oscillations in video quality* which can degrade a user's QoE.

Figure 11 compares the QoE-aware video delivery performed with several target video qualities against the DASH streaming and streaming of video in the existing resolutions (240p and 360p), in terms of video quality gains and bandwidth savings. This comparison was performed using the 4 minute long IronMan video and trace-based shaped bandwidth (with mean of 464 Kbytes/s and standard deviation of 157.51). Bandwidth was shaped using the traffic control (tc) command on Linux machine, using a mobile user data rate trace as input to periodically set a new maximum data rate value. We averaged each five seconds of the data rate trace into a single value, sending it as an input into traffic shaper every 5 seconds.

DASH streaming was executed in Google Chrome browser on the same machine as traffic shaper, while video chunks and the streamer resided on the server machine that was in the same LAN as the client. Streaming was programmatically invoked to start at the same time as traffic shaping. During streaming output data rates were recorded using Wireshark every second and saved for later use in QoE-aware streaming. DASH streaming was performed using the commercial Webm DASH player [27] due to being able to play videos encoded in webm format. QoE-aware video delivery was implemented in Matlab that enables quick evaluation of early ideas, while the full prototype is planned for future work. Streaming of video in fixed resolutions was emulated over the same channel in Matlab, verifying if there are any playback interruptions.

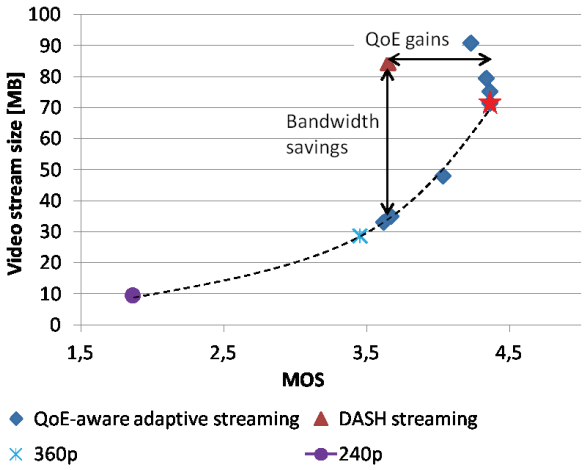

Fig. 11: Performance comparison of QoE-aware video delivery and DASH streaming

MOS was derived from VQM score using VQM/MOS mapping, while the VQM score of 4 minute video was computed as the 5% worst video quality of all 15 seconds video stream segments' VQM scores[7].

The optimal target video quality of QoE-aware video delivery is illustrated in Figure 11 with a red star, lying on the same dashed trendline as videos in fixed resolutions and other QoE-aware delivery points that were not degraded in quality due to lack of available bandwidth. Note that streaming video in the resolutions higher than 360p could not be performed without interruptions or quality degradations. Therefore, streaming a video in single resolution **limits QoE** that can be experienced over the given channel and **motivates the need for optimizing videos for different perceptual video qualities**.

Figure 11 also depicts several QoE-aware delivery points that deviate from the trendline, being optimized for higher

[6]The proposed QoE-aware video delivery can be also run online in a DASH-compliant video streamer by specifying optimized video segments in the target video quality in the highest representation of Media Presentation Description file, followed by the segments in lower qualities specified in the lower representations.

[7]Since there is no standardized method that can evaluate the perceived video quality of the video longer than 15 seconds, we plan as part of future work to verify using subjective quality tests on longer video clips if the users would give the same (or similar) video quality scores in reality.

video qualities than the bandwidth allows, which caused switching from optimized video streaming to DASH using lower bitrates and degrading perceptual video quality. Furthermore, observe that DASH point lies outside of trendline, with lower perceptual quality and higher number of bits than the optimized video stream for this channel requires. This indicates **potential gains of QoE-aware video delivery over DASH**.

The optimal target video quality is *typically experimentally found*, by streaming optimizing videos for different target qualities over an emulated channel and selecting the one that results in the highest MOS. This quality can also be *predicted before the streaming starts*, using the limited information that can be available on the user's device: average data rate of the channel and optimized video segments' bitrates. The details about this prediction method and performance of QoE-aware adaptive video streaming can be found in our other paper [7].

VIII. Conclusion

This paper proposes a method that optimizes video for different perceptual video qualities, reducing a video size and achieving potential bandwidth (and storage) savings for content providers, operators, and end users. This method uses DASH to split a video in segments of short duration that can be encoded in different resolutions and bitrates, aggregating these segments into a single video stream. Video optimization is performed by downscaling resolution of video segments, whose quality did not degrade much after downscaling, as determined by VQM.

Optimized video can be recorded into a file and played on a user's device, or can be streamed using DASH when there is enough available bandwidth or bits in the buffer (otherwise switching to adaptive streaming with lower resolutions/bitrates). In the former scenario, we envisage a multimedia file system downloading optimized videos according to individual user's perception to a mobile device, passing a QoE model to the server/cloud for video optimization. In the latter scenario, video is streamed to a user's device in optimal quality for the given conditions. This method can improve QoE of DASH streaming with fewer required bits.

The video optimization results were obtained with Samsung Galaxy S3 smartphone with maximum supported 720p resolution, therefore they cannot be applied to another device form factor. In order to be applicable in different environments, the described experiments and methodology need to be repeated with other devices, video types, and viewing environments, which is planned for future work. We will also investigate if grouping devices and videos allow reusing some of the work across devices and videos belonging to the same group. As environmental context also affects the mobile video QoE [13], it will be integrated into video optimization method.

References

[1] Advanced encoding settings : Youtube. www.support.google.com/youtube/bin/answer.py?hl=en&answer=1722171.

[2] FFmpeg. www.ffmpeg.org.

[3] Adobe. U.S. Digital Benchmark Adobe Digital Index Q2 2014. www.cmo.com/content/dam/CMO_Other/ADI/Video_Benchmark_Q2_2014/video_benchmark_report-2014.pdf, Oct. 2014.

[4] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2013-2018. www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf, June 2014.

[5] M. Claypool and A. Tripathi. Adaptive Video Streaming using Content-Aware Media Scaling. WPI-CS-TR-04-01, Computer Science Technical Report Series, Worcester Polytechnic Institute, 2004.

[6] Conviva. Viewer Experience Report. www.conviva.com/vxr/, Feb. 2013.

[7] A. Devlic, P. Kamaraju, P. Lungaro, Z. Segall, and K. Tollmar. Towards QoE-aware adaptive video streaming. In *Proc. of IWQoS*, Portland, Oregon, USA, June 2015.

[8] S. Egger, T. Hossfeld, R. Schatz, and M. Fiedler. Waiting Times in Quality of Experience for Web Based Services. In *Proc. of QoMEX*, pages 86–96, Melbourne, Australia, July 2012.

[9] Ericsson. Identifying the needs of future consumers. www.ericsson.com/news/130828-identifying-the-needs-of-tomorrows-video-consumers_244129227_c, Aug. 2013.

[10] B. Hankinson. Streaming 4K/8K Video over IP Networks: Daunting Problems, Proposed Solutions. https://www.streamonix.com/pdfs/streamonix_cinegrid_presentation.pdf, Dec. 2013.

[11] ISO/IEC DIS 23009-1.2. Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats, Apr. 2012.

[12] ITU-R. Methodology for the subjective assessment of the quality of television pictures. ITU-R Recommendation BT.500-13, Jan. 2012.

[13] S. Jumisko-Pyykk and M. M. Hannuksela. Does Context Matter in Quality Evaluation of Mobile Television? In *Proc. of MobileHCI*, pages 63–72, Amsterdam, Netherlands, Sept. 2008.

[14] M. S. Landy. Weber's Law and Fechner's Law, Handouts. www.cns.nyu.edu/~msl/courses/0044/handouts/Weber.pdf, Sept. 2014.

[15] Y.-S. Li, C.-C. Chen, T.-A. Lin, C.-H. Hsu, Y. Wang, and X. Liu. An end-to-end testbed for scalable video streaming to mobile devices over HTTP. In *Proc. of ICME*, pages 1–6, San Jose, CA, USA, July 2013.

[16] Z. Li, A. C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran. Streaming Video over HTTP with Consistant Quality. In *Proc. of MMSys*, pages 248–258, Singapore, Mar. 2014.

[17] D. Miras and G. Knight. Smooth Quality Streaming of Live Internet Video. In *Proc. of IEEE Globecom, Vol. 2*, pages 627–633, Dallas, Texas, USA, Dec. 2004.

[18] R. K. P. Mock, X. Luo, E. W. W. Chan, and R. K. C. Chang. QDASH: A QOE-aware DASH system. In *Proc. of MMSys*, pages 11–22, Chapel Hill, North Carolina, Feb. 2012.

[19] National Telecommunications & Information Administration (NTIA) Institute for Telecommunication Sciences (ITS). VQM software. www.its.bldrdoc.gov/resources/video-quality-research/software.aspx.

[20] M. Pinson and S. Wolf. A New Standardized Method for Objectively Measuring Video Quality. *IEEE Transactions on Broadcasting*, 50:312–322, Sept. 2004.

[21] P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo. The Logarithmic Nature of QoE and the Role of the Weber-Fechner Law in QoE Assessment. In *Proc. of IEEE International Conference on Communications (ICC)*, pages 1–5, Cape Town, South Africa, May 2010.

[22] J. Ruckert, O. Abboud, T. Zinner, R. Steinmetz, and D. Hausheer. Quality Adaptation in P2P Video Streaming Based on Objective QoE Metrics. In *IFIP Networking*, Prague, Czech Republic, May 2012.

[23] 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*, 17:1103–1120, Sept. 2007.

[24] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia. A Survey on Quality of Experience of HTTP Adaptive Streaming. *IEEE Communications Surveys & Tutorials*, PP, Sep. 2014.

[25] T. Virtanen et. al. Forming valid scales for subjective video quality measurement based on a hybrid qualitative/quantitative methodology. In *IS&T/SPIE Electronic Imaging*, San Jose, CA, USA, Jan. 2008.

[26] VQEG. Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality. ITU-T SG 9, Contribution COM 9-80-E, June 2000.

[27] Webm. Instructions to playback Adaptive WebM using DASH. wiki.webmproject.org/adaptive-streaming/instructions-to-playback-adaptive-webm-using-dash.

[28] S. Winkler. Digital Video Quality: Vision Models and Metrics. John Wiley & Sons, Ltd., 2005.

[29] S. Wolf and M. H. Pinson. Spatial-temporal distortion metrics for in-service quality monitoring of any digital video system. In *Proceedings of the SPIE 3845, Multimedia Systems and Applications II*, pages 266–277, Boston, Massachusets, USA, Sept. 1999.

[30] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld. Impact of Frame Rate and Resolution on Objective QoE Metrics. In *Proc. of QoMEX*, pages 29–34, Trondheim, Norway, June 2010.