App for session monitoring and subjective quality assessment of 2D and 360VR video in real time

Javier Escobar , César Díaz , Fernando Jaureguizar , Pablo Pérez and Narciso García

Abstract—We present an app to both monitor the session activity and conduct subjective quality tests of both 2D and 360VR content transmitted through adaptive bitrate streaming techniques in real time in an unsupervised, highly-configurable, minimum-intrusive, integrated fashion. Users carry out the tests within the app environment, thus fostering immersiveness and engagement. Furthermore, monitoring and test results are collected and stored in a cloud database for subsequent analysis.

Index Terms—Adaptive streaming, quality of experience, quality monitoring, quality assessment, real time, 360VR video

I. INTRODUCTION

Service and content providers and test designers have always aimed for tools to monitor and assess video quality in a distributed, sufficiently-automatic, little-intrusive and unsupervised manner, with the goal of obtaining more results and more reliable, considering that the environment where the test is conducted heavily conditions the perception of the subjects [1]. Furthermore, recent advances in the area of Virtual Reality (VR), specifically regarding the streaming and visualization of 360 video content (360VR) on head-mounted displays (HMD) are heavily fostering the need for measurements, procedures and tools to properly measure and assess the quality perceived by users in these environments [2], [3]. In this scenario, immersiveness and engagement are key to how users perceive the presented content and need to be observed by the strategies and tools used to conduct the evaluations [4].

Considering the previous elements, we have designed and implemented an app that is capable of monitoring multimedia quality and performing synchronous subjective video quality tests in real time in a highly-automatic, minimum-intrusive and unsupervised manner. This app considers content encoded and segmented to be delivered using HTTP adaptive bitrate streaming (ABR) technology, and so it is perfectly suited to study the impact of the channel dynamics on the video quality perceived by users [5], [6]. This user-friendly app allows self-guided operation of the tool, not requiring any in-test assistance. In addition, all the data that results from the conducted tests and the quality monitoring are collected and sent to a cloud database for real time or subsequent analysis.

II. PROCEDURE

The course of action followed by our app is depicted in Fig. 1. First, the test designer creates offline all the tests to be conducted. The rest of the steps are carried out by the subject online as follows. Once the app is opened, a welcome form pops up. The user just needs to answer a couple of quick questions, select a configuration file and press the *Start* button. This leads to the playback of a number of segments in a row at potentially different qualities (specified in the configuration file). After these segments have been played, a test screen shows up. This new environment involves one or more questions that are answered subsequently by the user. After the evaluation of the first period of content is completed, the playback is resumed. These steps are repeated until the session is over. The subject is not required to interact with anybody or, in the case of tests of 360VR content, to remove the HMD at any moment throughout the session.

In addition, during the presentation of the content, the app monitors periodically several variables that determine the behavior of the user during a given period: the location where he/she is looking at and three metrics that are key to infer the quality of experience (QoE) of the user: the average quality of the played segments in terms of bitrate, Q , the weighted average of the switches between qualities, QS , and the accumulated duration of all service stalls, SS . Respectively, these variables are computed as:

$$Q = \frac{1}{N_{\text{seg}}} \cdot \sum_{s=1}^{N_{\text{seg}}} R(s) \quad (1)$$

$$QS = \frac{1}{N_{\text{seg}}} \cdot \sum_{s=1}^{N_{\text{seg}}-1} QS(s \rightarrow s+1) \quad (2)$$

where:

$$QS(s \rightarrow s+1) = \begin{cases} \alpha \cdot \frac{R(s+1)-R(s)}{R(s)} & : R(s+1) \geq R(s) \\ (1-\alpha) \cdot \frac{R(s+1)-R(s)}{R(s+1)} & : R(s+1) < R(s) \end{cases} \quad (3)$$

$$SS = \frac{1}{N_{\text{stall}}} \cdot \sum_{u=1}^{N_{\text{stall}}} D(u) \quad (4)$$

Fig. 1. Course of action of the presented application program

where $R(s)$ is the bitrate of segment s , N_{seg} the number of segments in the period, $QS(s \rightarrow s+1)$ the depth of the switch between segments s and $s+1$, α the weight given to upward switches, as switches are perceived unequally not only qualitatively (upward ones are good, downward ones are bad), but also quantitatively [5], N_{stall} the number of stalls, and $D(u)$ the duration of stall u .

Finally, the test designer can have immediate feedback of the process, as the obtained data are collected and stored in a cloud database. Furthermore, these actions will result in a huge, constantly-updating database that can potentially include data of thousands of sessions. The data stored in the cloud database can be analyzed in a more or less segmented fashion depending on the specific needs of the miner.

III. ARCHITECTURE

The app is actually an integrated environment made up of several interrelated subapps that are briefly described next.

The *Test configuration app* is used by test designers to fully configure offline the tests to be conducted. Test designers can create any number of configuration files. Within each configuration file, they can select the content and qualities to be shown to the subject throughout the session. Furthermore, they can include any Mean Opinion Score (MOS)-oriented test method (ACR, DCR...) [7], [8], [9] or yes-no questions. Also, they can set the time between consecutive tests. The actual file to be used in a particular test can be selected either by the test designer, by the user or randomly by the core app.

The *Welcome app* includes instructions that the subject needs to know before starting the test, as well as the form used to collect any data required to perform a segmented analysis (gender, age, location...). It also includes drop-down lists to select configuration files created by the test designer using the Test configuration app described above. Finally, it includes a *Start* button to make sure that the session does not start unexpectedly before the subject is ready.

The *Core app* defines the main behavior of the program. It launches the necessary activities for the session. It reads the selected configuration file and sets the whole system to work accordingly and the type of content used: 2D or 360VR. It launches the adapted player app and sends all the monitoring data (periodically every few hundreds of milliseconds) and assessment data (any time the subject finishes any partial evaluations) to the server that hosts the selected database.

The *Adapted player app* background functioning is that of an ABR client. However, its basic behavior has been modified to manage all segment requests considering the configuration file. To properly carry out a given configured subjective test, we might need to use a simulated environment, where any quality and/or content variations can be requested and played out without any stalls. To that end, it is necessary for the subject to guarantee that network conditions will always be good enough to properly support all tested video qualities. Lastly, the received multimedia segments are delivered along to the media decoder and player.

The *VR app* is used in the case of showing 360VR content to set the environment to interact with the adapted player app and render the video content and the 360VR-adapted test screens.

The *Test app* is launched every time a period for content playback is over. When this app is open, a series of subsequent test screens pop up, each of them presenting a different question that the user is requested to answer. In most tests, only one question will be posed per period. The subject may take all the time he/she needs to answer each question (response times are not predetermined).

REFERENCES

- [1] T. Hoßfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold, and P. Tran-Gia, "Best practices for QoE crowdtesting: QoE assessment with crowdsourcing," *IEEE Trans. Multimedia*, vol. 16, pp. 541–558, Feb. 2014.
- [2] L. Brun and A. Gasparini, "Enabling 360 visual communications: Next-level applications and connections," *IEEE Consum. Electron. Mag.*, vol. 5, no. 2, pp. 38–43, Apr. 2016.
- [3] L. Turban, F. Urban, and P. Guillotel, "Extrafoveal Video Extension for an Immersive Viewing Experience," *IEEE Trans. on Visualization and Computer Graph.*, vol. 23, no. 5, pp. 1520–1533, May 2017.
- [4] C. Díaz, J. Cabrera, M. Orduna, L. Muñoz, P. Pérez, J. Ruiz and N. García, "Viability analysis of content preparation configurations to deliver 360VR video via MPEG-DASH technology," *IEEE Int. Conf. on Consum. Electron. (ICCE 2018)*, pp. 1–2, Jan. 2018.
- [5] M.-N. García, F. De Simone, S. Tavakoli, N. Staelens, S. Egger and K. Brunnström and A. Raake, "Quality of experience and HTTP adaptive streaming: a review of subjective studies," *Int. Workshop on Quality of Multimedia Experience (QoMEX 2014)*, pp. 141–146, Sep. 2014.
- [6] P. Pérez, J. Macías, J.J. Ruiz and N. García, "Effect of Packet Loss in Video Quality of Experience," *Bell Labs Tech.J.*, vol. 16, no. 1, pp. 91–104, Jun. 2011.
- [7] Recommendation ITU-T P.910, "Subjective video quality assessment methods for multimedia applications," Apr. 2008.
- [8] Recommendation ITU-R BT.500-13, "Methodology for the subjective assessment of the quality of television pictures," Jan. 2012.
- [9] Recommendation ITU-T P.913, "Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment," Mar. 2016.