University of Würzburg
Institute of Computer Science
Research Report Series

Best Practices for QoE Crowdtesting: QoE Assessment with Crowdsourcing

Tobias Hoßfeld, Christian Keimel, Matthias Hirth,
Bruno Gardlo, Julian Habigt, Klaus Diepold,
Phuoc Tran-Gia

Report No. 486 October 2013

[1] University of Würzburg, Institute of Computer Science
Am Hubland, D-97074 Würzburg, Germany
hossfeld@informatik.uni-wuerzburg.de

[2] Institute for Data Processing, Technische Universität München
Arcisstr. 21, 80333 München, Germany
christian.keimel@tum.de

[3] Telecommunications Research Center Vienna - FTW
Donau-City-Straße 1, A-1220 Vienna, Austria
gardlo@ftw.at

Acknowledgments. This work was partly funded by Deutsche Forschungsgemeinschaft (DFG) under grants HO 4770/1-1 and TR257/31-1, and the COST QUALINET Action IC1003. The authors alone are responsible for the content.

Best Practices for QoE Crowdtesting: QoE Assessment with Crowdsourcing

Tobias Hoßfeld, Matthias Hirth, Phuoc Tran-Gia
University of Würzburg, Institute of Computer Science
Am Hubland, D-97074 Würzburg, Germany
hossfeld@informatik.uni-wuerzburg.de

Christian Keimel, Julian Habigt, Klaus Diepold
Institute for Data Processing, Technische Universität München
Arcisstr. 21, 80333 München, Germany
christian.keimel@tum.de

Bruno Gardlo
Telecommunications Research Center Vienna - FTW
Donau-City-Straße 1, A-1220 Vienna, Austria
gardlo@ftw.at

Abstract

Quality of Experience (QoE) in multimedia applications is closely linked to the end users' perception and therefore its assessment requires subjective user studies in order to evaluate the degree of delight or annoyance as experienced by the users. *QoE crowdtesting* refers to QoE assessment using crowdsourcing, where anonymous test subjects conduct subjective tests remotely in their preferred environment. The advantages of QoE crowdtesting lie not only in the reduced time and costs for the tests, but also in a large and diverse panel of international, geographically distributed users in realistic user settings. However, conceptual and technical challenges emerge due to the remote test settings. Key issues arising from QoE crowdtesting include the reliability of user ratings, the influence of incentives, payment schemes and the unknown environmental context of the tests on the results. In order to counter these issues, strategies and methods need to be developed, included in the test design, and also implemented in the actual test campaign, while statistical methods are required to identify reliable user ratings and to ensure high data quality. This contribution therefore provides a collection of best practices addressing these issues based on our experience gained in a large set of conducted QoE crowdtesting studies. The focus of this article is in particular on the issue of reliability and we use video quality assessment as an example for the proposed best practices, showing that our recommended *two-stage QoE crowdtesting design* leads to more reliable results.

1 Introduction

Subjective testing is an integral part of the research on multimedia technology and algorithms, as any new concept needs to be validated with respect to the suitability for the potential users. Besides usability, acceptability and task performance, the users' overall *Quality of Experience (QoE)* in the context of multimedia applications is often a focus of subjective tests. Although for some areas objective metrics exist that can replace subjective QoE testing, such metrics are often limited to well-defined subsets of possible application scenarios and therefore not universally applicable, leading to unreliable results for scenarios not considered during the design of such metrics. Hence, subjective QoE assessment tests are still needed in the research on multimedia topics. These tests,

however, are expensive from both an organisational and a financial perspective: test subjects need to be recruited and test sessions need to be organised, often with constraints on the number of test subjects that can participate simultaneously in the laboratory, leading to time consuming test campaigns and a lack of flexibility. Furthermore, due to the fixed location of the laboratory, the subjects may not be a representative sample of the complete population in a statistical sense. Additionally, test subjects often need to be reimbursed on a competitive wage level in order to get a sufficient number of test subjects. Therefore subjective testing can often strain the available resources, resulting in either a compromise in the number of considered test cases or avoiding the subjective testing altogether.

QoE crowdtesting provides an alternative to the traditional subjective testing, aiming at reducing the resources necessary for conducting subjective testing by utilising *crowdsourcing*. Crowdsourcing is a relatively new concept that outsources tasks via the Internet to a global worker pool, resulting in reduced costs, larger diversity of the test subjects, and faster turnover of test campaigns. Even though the use of the Internet as a virtual laboratory leads to limitations on the stimuli and scenarios that can be tested, the ever increasing bandwidth and capabilities of the connected devices allow for a wide range of areas in which QoE crowdtesting can be used. QoE crowdtesting, however, is not just a straight forward implementation of existing subjective testing methodologies in an Internet-based environment. Owing to the fundamental differences between the traditional and virtual laboratory, extra considerations need to be taken in order to gain reliable results.

In this contribution, we therefore provide a collection of best practices for QoE crowdtesting by addressing on the one hand the key issues that need to be considered if a subjective test should be replaced by QoE crowdtesting, and, on the other hand, how these issues can be addressed best in the design and implementation of the desired QoE crowdtesting campaign. In particular, how the participating test subjects can be screened with regards to their reliability. As unlike in a traditional laboratory environment, the virtual laboratory provided by QoE crowdtesting does usually not allow for subject-supervisor interactions. We have chosen the QoE assessment of video as an example to illustrate the proposed best practices, but the presented best practices can also be applied to the QoE assessment of other stimuli.

The remainder of this contribution is structured as follows. Section 2 gives a background on crowdsourcing and the differences of common crowdsourcing platforms and provides an introduction in the use of the crowdsourcing principle to assess QoE by means of subjective user studies with QoE crowdtesting. The key issues of QoE crowdtesting are summarized in Section 3 addressing limitations, reliability, incentives and task design, context monitoring, and hidden influence factors. Technical challenges and best practices for the implementation of QoE crowdtesting are analysed in Section 4. The statistical analysis of the obtained user ratings from QoE crowdtesting is shown in Section 5, where we show the need and mechanisms for filtering out unreliable user ratings. Based on the preceding sections, we then present in Section 6 the proposed best practices. Finally, Section 7 summarizes this work and gives an outlook on important future steps for QoE crowdtesting.

2 Background on Crowdsourcing and QoE

An overview of the general principles of crowdsourcing is given and the crowdsourcing related terminology used in the remainder of this article is introduced (Section 2.1). Further we highlight the differences of existing crowdsourcing platform types and their strengths and weaknesses for QoE crowdtesting (Section 2.2). Since QoE assessment is our target use case in the crowdsourcing platform, QoE and its influence factors are discussed first (Section 2.3), before QoE crowdtesting in general and in particular for video QoE assessment is discussed (Section 2.4).

2.1 Principle of Crowdsourcing

Crowdsourcing can be considered as a further development of the outsourcing principle, where *tasks* are submitted to a huge crowd of anonymous *workers* in the form of an open call, instead of a designated employee or subcontractor [1]. Crowdsourcing tasks in general are very diverse and can range from simple word recognition [2] to complex research and development tasks [3]. The granularity of these crowdsourcing tasks, however, differs from the granularity of tasks in traditional forms of work [4], as workers usually do not have the same background knowledge and overview of the context of a task compared to a full time employee. Therefore, the crowdsourcing tasks are typically small and atomic.

QoE crowdtesting belongs to the category of *micro-tasks*. Micro-tasks can usually be accomplished within a few minutes to a few hours and do not require a long-term employment. They are often highly repetitive, for instance generating consecutive measurement samples, and are usually grouped in larger units, which we refer to as *campaigns*. For conciseness we will from now on not differentiate between tasks and micro-tasks. Most *employers* submitting tasks to an anonymous crowd use mediators which maintain the crowd and manage the employers' campaigns. These mediators are called crowdsourcing platforms. Crowdsourcing platforms in general can roughly be distinguished into three different types, aggregator platforms, specialized platforms and crowd provider platforms. These platform types focus on the decomposition of larger tasks into crowdsourcable tasks, enable access to small specialized crowds, such as workers with certain devices, or provide a huge and diverse workforce.

2.2 Differences of Crowdsourcing Platform Types

Mediator crowdsourcing platforms, specialized crowdsourcing platforms, and platforms focusing on crowd provision differ among each other in terms of their capabilities and main use cases. This results in individual advantages and drawbacks of the platform types in the context of QoE crowdtesting. Figure 1 illustrates the types of crowdsourcing platforms and their interactions.

Aggregator platforms can be seen as the most high-level type of crowdsourcing platforms. They often do not maintain an own workforce but recruit workers from different channels, like specialized platforms or crowd provider platforms. The main business case

of these platforms is the development of crowd-based solutions for existing work flows which are not crowdsourced, yet. Therefore, the targeted employers of these platforms are usually companies trying to integrate crowdsourcing in their daily business. Besides this, aggregator platforms also offer self-service for smaller employers. Here, the aggregator platforms often focus on a specific subset of tasks for which they also offer predefined quality assurance mechanisms.

The advantage of this platform type is the high abstraction of the crowdsourcing related issues, like worker recruiting or quality control. Usually only the required number of submitted tasks has to be defined, the recruiting process is automated by the platform. On some platforms it is even possible to adjust the data quality via a simple slider on the platforms' web interface. However, the underlying quality mechanisms are mainly optimized for simple tasks, like image tagging.

The high abstraction of these platforms is also their major drawback with regard to QoE crowdtesting. Due to platform internal recruiting mechanisms, the available workers might already be pre-filtered, which limits their diversity. Furthermore, the available quality assurance methods are usually not applicable for the quality control of QoE crowdtesting tasks. Therefore, still additional monitoring of the users is required. Aggregator platforms also add an additional business layer between the employer and the worker, which also increases the costs per task. Currently available aggregator platforms are for example Crowdflower [5] or Crowdsource [6].

Similar to aggregator platforms, *specialized crowdsourcing platforms* only focus on a limited subset of tasks or on a certain type of worker. In contrast, specialized crowdsourcing platforms have their own work force. With regard to QoE crowdtesting, specialized platforms focusing on specific tasks, e.g. Microtask [7], have similar advantages and disadvantages as aggregator platforms. Due to the task specialization and self-service QoE crowdtesting campaigns might not be possible at all. In contrast, the use of crowdsourcing platforms which focus on a specific set of workers is useful if only a limited subset of workers, for example from a given location or with a specific mobile device, is requited [8].

The most flexible type of crowdsourcing platform are *crowd providers*, like Amazon Mechanical Turk (MTurk) [9] or Microworkers [10]. These platforms focus mainly on self-service and maintaining a huge worker crowd. This crowd can be directly accessed through the web interface of the platform or via an API for automatic interactions. Commercial crowd providers often implement a set of filters and qualification mechanisms to select and build specialized worker groups. Due to the direct access to the crowd workers, crowd providers offer the largest flexibility in terms of task and campaign design. These platforms also accumulate a vast unfiltered number of workers from all over the world, which results in a large diversity of the potential testers.

However, due to the variety of the tasks on this type of platform, the operators usually only provide a very limited set of quality assurance mechanisms and therefore advanced mechanisms must be integrated by the employer into the tasks in this case.

Besides commercial crowd providers, Facebook [11] and other social networks can be used to recruit test users as well. If a user test can be implemented in a joyful manner, social networks allow to easily reach a large number of test subjects for free. The test

can sometimes also be integrated in a Facebook app which additionally enables access to the users' demographic information provided in their profiles. Redesigning a user test to be joyful and integrating it in a Facebook app, however, imposes a significant amount of additional work and is not always possible. Furthermore, participants recruited from a social network might be biased in terms of expectations of test behaviour, if they are familiar with the creator of the test of belonging to the same community.

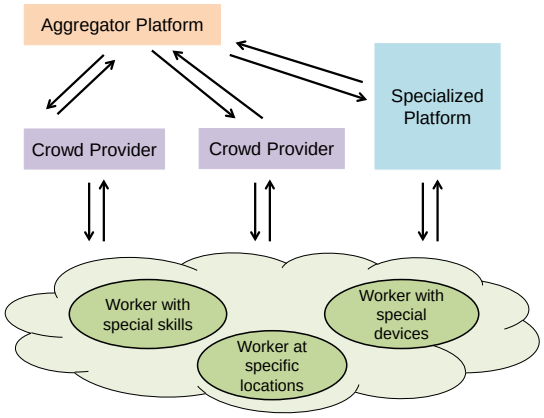


Figure 1: Types of crowdsourcing platforms and their interactions.

2.3 Quality of Experience

One possible definition of QoE in the context of multimedia systems and applications is provided in [12] as "the degree of delight or annoyance of the user of an application or service. It results from the fulfilment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the users personality and current state". In this definition, QoE is influenced by a variety of factors [12, 13] that can be divided into four different categories, each representing a different level in multimedia systems and applications: *context, user, system*, and *content* level. The *context level* considers aspects like the environment in which the user is consuming the service, the user's social and cultural background or the purpose of using the service, for example recreation or information retrieval. The *user level* includes psychological factors like expectations of the user, memory and recency effects or the usage history of the application. The technical influence factors are abstracted on the *system level*. They cover influences of the transmission network, the devices and screens, but also of the implementation of the application itself like, video buffering strategies. Lastly, the *content level* addresses characteristics of the content, for example for video, the video codec, format, resolution, but also duration, content of the video, type of video and its motion patterns.

Besides this more general discussion, QoE can also be considered with a focus on certain applications, for example, cloud applications [14], mobile applications [15] or scalable video delivery [16], and, more generally, video, voice, and web services [17].

2.4 QoE Crowdtesting with Emphasis on Video Applications

The aim of QoE crowdtesting is to move the QoE assessment from a standardized lab environment into the Internet, where the crowdsourcing platforms act as an extra layer between test manager and test subject, handling the recruiting and payment of the test participants. The subjective testing is therefore using subjects from a global worker pool, usually with a web-based application, that can be accessed via common web browsers as Firefox, Internet Explorer, or Google Chrome.

Video QoE assessment is done for a range of different application areas: from the visual quality evaluation of video coding technologies and processing algorithms to the influence of network delays and packet loss on the video quality. The QoE of video is usually determined in a well-defined testing environment with subjective methodologies, as described in standards like [18, 19]. In the context of QoE crowdtesting, we must distinguish between two categories of video QoE assessment: QoE evaluation of Internet-based video applications for instance YouTube and QoE assessment of video in general such as the evaluation of coding technologies. The difference between these two categories lies in the fact that the Internet-based video applications are already by their very nature optimized for the presentation in a web environment and can therefore be easily adapted to QoE crowdtesting. In contrast, applying crowdtesting to video QoE assessment in general necessitates the additional design of Internet-based applications for the presentation of the videos under test. Both categories will be discussed briefly in this section.

QoE crowdtesting of Internet-based video applications is relatively straightforward, as the main difference to the lab is the use of crowdsourcing platforms for test subject recruitment and reimbursement. Although some adaptations for interfacing with the crowdsourcing platforms may be necessary, the application itself needs not to be modified. One typical example of this category is the influence of stalling events in video streaming on the video QoE as discussed, for example, for YouTube in [20]. Here, the test setup in the lab usually already consists of a web interface presenting the videos and collecting the subjects' scores. However, in order to avoid additional stalling caused by the test users' Internet connection, the videos had to be downloaded completely to the browser cache before playing. During the initial download of the videos, a personal data questionnaire may be completed by the participant including also consistency questions to check for reliability [20].

For general video QoE assessment, the adaptation of the lab tests to QoE crowdtesting is more comprehensive. Firstly, the testing methodology needs to be provided with an Internet application, instead of platform-dependent software. Secondly, the delivery of the videos under test must be implemented. Especially for testing methodologies, that are based on an uncompressed video for comparison, this requires dedicated applications. Alternatively, a video crowdtesting platform like *QualityCrowd* [21] can be used, that already takes these issues into consideration. In addition, it may also be necessary to adapt the goal of the test to the limitations of the crowdsourcing environment. For example, videos with spatial or temporal resolution beyond the capabilities of consumer equipment need to be down-sampled.

Common to both categories is, that instead of a sophisticated hardware and standardized test environment, the hardware and viewing environment will vary between the different workers. This *lack of control* can be tackled with the different strategies of *monitoring*, *adaptation* and *prevention* as will be discussed in detail in Section 3.5. In contrast to these environmental issues, however, common subjective testing methodologies for video quality assessment can be used. Using ITU-R BT.500 [22], for example, both the discrete double stimulus DSIS and the continuous double stimulus DSCQE methodologies can be implemented easily in a corresponding web interface.

Studies from literature have shown that using crowdtesting for the QoE assessment of a wide range of video applications can deliver results similar to traditional testing in the lab environment: Keimel et al. have shown in [23, 21] that crowdtesting delivers results within the acceptable inter-lab variation between different testing labs for standard conforming QoE assessment, Chen et al. discussed crowdtesting for audio-visual QoE of Internet-based applications in [24, 25], which was discussed more in detail by Wu et al. in [26], and Hossfeld et al. applied crowdtesting to the influence of stalling events [20] and initial delays [27] on the QoE in video streaming applications. For pairwise comparison QoE tests, Xu et al. suggest an approach to decompose the pairwise comparison data onto random graphs in [28], reducing the assessment tasks for each participant significantly and therefore making pairwise comparison more suitable for crowdtesting.

3 Key Issues in QoE Crowdtesting

With crowdsourcing, researchers have new possibilities to conduct subjective user studies. For QoE assessment, however, conceptual challenges arise by moving the subjective user studies to the crowd due to the typically short micro-tasks compared to long lab studies (Section 3.2). Additional challenges emerge due to the remote setting of the test users as well as the heterogeneity of users, used hardware, environment settings, etc. On one hand, the actual user ratings are affected because of the QoE influence factors which are additionally emerging from the remote setting and which are not directly controlled. On the other hand, the execution of the test study and the implementation of the (web-based) test software has to consider the crowdsourcing settings and the non-standard test equipment, e.g. software compatibility to ensure a successful execution of the test, e.g. Internet access speed for downloading the test contents which may result into undesired waiting times during the subjective study. The emerging challenges are the reliability of users and user ratings (Section 3.3), incentives and task design for attracting test users (Section 3.4), unknown context of users in tests and other hidden influence factors like diverging expectations of users (Section 3.5). Beside the QoE aspects to be tested, sophisticated mechanisms have to be developed, included in the test design, and implemented in the actual test campaign to cope with those key issues.

3.1 Limitations of QoE Crowdtesting

In principle, QoE crowdtesting could be used for the assessment of any stimuli and interactivity, using any type of subjective methodology. In reality, however, we are faced

with several limitations on the possible scope of QoE crowdtesting.

The main technical factors limiting the scope of QoE assessment are bandwidth constraints and support of the workers' devices to present the required stimuli. The first factor requires to consider the support of coding standards by the workers devices, as it is often not feasible to provide the uncompressed stimuli to the workers due to excessive bandwidth demands. This is in contrast to the traditional lab setting, where the aim is to avoid any additional compression of the stimuli under test. But even with supported codecs, the size of compressed stimuli may be too large for the connection bandwidth of many workers, especially for HDTV or even UHDTV formats.

Secondly, the stimuli must be supported by the workers' devices. Although 2-D video and audio capabilities have become standard at most devices, 3-D video and audio capabilities or high dynamic range (HDR) displays cannot be readily assumed to be available. The support for other stimuli, for example, haptic or olfactory stimuli, is nearly non-existent in common computer hardware as used by the workers and thus these stimuli are currently not suitable for QoE crowdtesting.

Besides these technical factors, QoE assessment methodologies requiring the interaction between different workers, e.g. for interactive video conferencing, are possible, but challenging in their execution.

Taking these limitations into account, QoE crowdtesting is feasible for 2-D video, image and audio QoE assessment tasks, where the usable formats depend on the bandwidth requirement. In particular, for video, HDTV formats, depending on the required bitrate, may be not suitable for QoE crowdtesting with today's Internet access speed.

3.2 Conceptual Challenges for QoE Crowdtesting

The migration to crowdsourcing invokes some *conceptual challenges* on how to assess QoE and how to design the user tests [29]. In laboratory studies user tests may take up to 90 minutes [30] which allow, for example, to investigate memory effects [31]. In contrast, crowdsourcing tasks are typically rather short and take at most a few minutes [32]. Therefore, tests designed for a lab environment need to be modified for crowdtesting and one of simplest way to this is by partitioning the test into basic test cells [29]. As a consequence, a crowdsourced QoE test user may only see a subset of the test conditions which requires sophisticated statistical methods for outlier detection or quantifying reliability. Another issue with QoE crowdtesting is the lack of a test moderator, but the user is guided via the web interface through the tests. In particular, the training of subjects is different than in a traditional lab environment and is mostly conducted by means of qualification tests. Nevertheless, in case of any problems with understanding the test, uncertainty about rating scales, sloppy execution of the test, or fatigue of the test user, appropriate mechanisms or statistical methods have to be applied.

3.3 Unreliability of Users: Reasons and Task Design Solutions

There are several reasons why some user ratings are not reliable and need to be filtered out. *Technical errors* may occur due to errors in the web-based test application or

due to incompatibilities of the test application with the worker's hard- and software including missing video codecs or insufficient screen resolution. As a consequence, the users observe different test conditions or additional artefacts occur leading to test results which appear unreliable, but may be valid for the individual users' conditions. This requires an appropriate monitoring of the system, but also of the context. Another possible reason for unreliable user ratings are the *test instructions* which may be not clear or too complex to understand, and additionally *language problems* may also occur with international users.

Furthermore, there may also be *cheating users*. Commercial crowdsourcing applications suffer from workers, who try to maximize their received payment while minimizing their own effort and therefore submit low quality work to obtain such a goal. To be more precise, the actual goal is the payment to effort ratio, and therefore tasks should be designed that incentivizes high quality work and not low quality work, as discussed later in Section 3.4. The submission of low quality work is the case, even if the expected gain is very little [33]. Thus, numerous efforts have been made in order to improve the quality of the results submitted by the workers and to detect cheating workers. The easiest way to test the trustworthiness and quality of a worker is to add *gold standard data* [34], where the correct task result is already known. Gold standard data can increase the quality of the task results as the worker receives an immediate feedback about mistakes and continuously cheating workers are easy to identify. In some cases gold standard data can be generated automatically [35] or even the bias of the workers can be taken into account [36]. Gold standard data is not applicable for tasks, however, where there is no clearly correct result, like in subjective rating. Here *content questions* and *consistency questions* can be used to estimate the reliability of a worker. In [37], Kittur et al. used crowdsourcing workers to rate the quality of Wikipedia articles. The correlation between the rating obtained from crowdsourcing and a trusted reference group was significantly improved by adding questions which test whether the worker read the article. Hossfeld et al. [20] also used content and consistency questions, but also added *application usage monitoring* for YouTube QoE tests. The most common approach is measuring the time the worker spends on the task. If the worker completes a task very quickly, this might indicate that the work was done sloppily. Also browser events (for web-based crowdsourcing tests) can be monitored in order to measure the focus time, which is the time interval during which the browser focus is on the website belonging to the user test. In order to increase the number of valid results from crowdsourcing, a warning message may be displayed. The users could decide to watch the video again or to continue the test. As a result of [20], this task specific user monitoring allows detecting unreliable subjects and the warning message doubled the number of reliable user ratings.

In [38], von Ahn and Dabbish present a crowd-based image labeling game which was used in an adapted version by Google's Image Labeler. A label is added to the picture, if at least two randomly picked users suggest the same label. Von Ahn and Dabbish argue that cheating is not possible due to the huge number of players. Two random players are very unlikely to know each other and, hence, are not able to collaborate. Besides the task design, the task type can also influence trustworthiness of the workers. Eickhoff and De Vries [39] observed that depending on the type of task more or less malicious workers are

encountered and suggested to derive the quality of a worker not only from the number of completed tasks but also their type, i.e. does the worker only perform simple tasks or mainly complex ones. Furthermore, the complete workflow of a crowdsourcing project can be optimized in order to detect cheaters and to improve the quality. Dow *et al.* [40] suggest to integrate an interactive feedback system to encourage workers and other contributions suggest to use multiple iterative tasks [41, 42] or coordination techniques [43] to improve the quality of the results. Hirth *et al.* [44] propose two generic crowd-based methods to ensure data quality with respect to the amount of costs, where gold standard data is not applicable and manual re-checking by the employer is ineffective. In particular, a majority decision approach and a control group approach are presented. A cost model for both approaches is developed in [44]. Using this cost model the main cost factors of both approaches were identified, and how the quality of the workers influences the weight of the different cost factors. The cost analysis also revealed that the majority decision approach is more suitable for low paid routine tasks, whereas the control group approach performs better for high priced tasks like QoE tests. The work in [44] shows that crowd-based cheat-detection mechanisms and quality control are cheap, reliable, and easy to implement.

Another relevant aspect of data quality is achieved by the concrete *task design*. Tests should be designed in such a way that there is no incentive for the user to cheat. Kittur *et al.* [37] conclude that in a task cheating should take approximately the same time as completing it properly and [45] discourages cheaters instead of detecting them by appropriate task design. Tasks that require creativity or abstract thinking decrease the ratio of cheaters in contrast to entertainment-driven workers, as money-driven workers prefer simple tasks over creative ones. Furthermore, long tasks should be split into smaller tasks, as the task duration has a severe impact on the cheater rate. Workers' share of previously accepted submissions provided by the platform, however, is not a robust measure of worker reliability [45].

3.4 Incentives and Payment Schemes

Incentives play a key role in the successful use of crowdsourcing in general and QoE crowdtesting in particular. Incentive design addresses the development of mechanisms and presentation of the task according to the following two goals: on one hand, incentive design aims to improve the willingness of subjects to participate beyond purely monetary interests, e.g. through gamification, and thus more users are completing the study in a shorter time. On the other hand, incentive design aims to improve the quality of the results generated by the subjects with incentive mechanisms that are complementary to reliability mechanisms [20] or data quality mechanisms [36].

While reliability mechanisms aim at filtering out unreliable users or unreliable results, data quality mechanisms try to estimate the quality of the workers or their submitted results in order to reject or block the low-performing workers. Different mechanisms for different domains have been proposed in literature: from image labelling [36] to natural language processing [46]. However, in the context of incentive design only a few insights and general conclusions are available. [47] shows that incentives encourage participants

to make more accurate judgements when using crowdsourcing for screening a number of candidates applying for a job at a company and to conduct resume reviews. Positive incentives were represented by bonus payments: each participant was initially told that each resume had already been rated by an expert and, if the participant's rating matched the expert's, the bonus was paid. In contrast, negative incentives were represented by telling the participant that their payment is reduced, if it differs from the expert's rating. A combination of positive and negative incentives was also applied. All incentive schemes in [47] increased the quality of work. Other payment schemes may depend on the actual performance of the worker. For example, the user is allowed to "choose as many as they want" test sequences for QoE assessment, and then they are paid accordingly to the number of evaluated tests sequences.

Beyond payment schemes, other incentives address social aspects, entertainment and altruism [48]. Altruistic crowdsourcing is carried out by volunteers with a desire to help, for example, in scientific research. Gamification or games with a purpose [49] is an approach to develop incentives for entertainment and fun, enabling human contributors to carry out computation tasks as a side effect of playing online games [46]. In the context of data or image labelling, different games are discussed in [50, 38]. However, there are no general guidelines how to design a game, as this is strongly task related. Nevertheless, the results for gamification of tasks are very promising: Eichhoff *et al.* [50] show that 70 % of users played more than the first round as necessary for payment and playing a second round does not bring any additional payments to the worker and thus the additional results are obtained for free by the employer. 80 % of the users return to a game, compared to only 23 % for a regular task and unreliable ratings in their task annotation game are reduced to 2.3 % instead of 13.5 %, compared to a non-gaming task. Innovative, creative tasks are less likely to be cheated on and also the time and cost is spent more efficiently. The quality of the results increased by 10 %. Thus, gamification has the potential to make crowdsourcing an even more powerful tool for QoE assessment.

3.5 Context Monitoring and Hidden Influence Factors

Because of the remoteness of the participants and the heterogeneity of the used soft- and hardware, it is necessary to monitor the users' environment in order to identify additional influence factors on the QoE assessment.

Influence factors are defined as any characteristic of a user, system, or context that may have influence on the users' QoE [12], where human influence factors are variant and invariant characteristics of a human user describing the demographic and socio-economic background, the physical and mental constitution, or the user's emotional state, and system influence factors are related to the media coding, transmission, storage, rendering, and reproduction/display. The context influence factors describe characteristics of the users' current environment that may influence the QoE. Due to the unknown context in which the QoE assessment is performed by the workers in QoE crowdtesting, these influence factors are not known beforehand, but hidden, yet still influence the users' QoE ratings.

In general, we have three options to cope with the unknown context and the resulting

hidden influence factors. We can either monitor the appropriate context parameters, adapt the context or try to prevent the undesirable context itself in our test design. In the following, we highlight some examples for best practices.

3.5.1 Monitoring of the Workers' Environmental Conditions and Context

The environment in which the workers evaluate the stimuli in QoE crowdtesting may impact the overall QoE and thus the application should be able to detect such factors. For visual stimuli, for example, the general viewing conditions represented by the background illumination or the screen resolution itself can be influencing factors.

One option to adapt the conditions of the workers' environment is to provide them with simple test patterns that allow them to either calibrate their devices or enable the quantification of the deviation of a device's stimuli representation from the desired target. For visual stimuli, a basic test pattern similar to the test patterns used for calibration of the monitor contrast and illumination in a professional environment can be utilised to quantify the users' viewing conditions, for example by asking how many grey steps on a greyscale step-wedge are visible. Moreover, such patterns can also be used to instruct workers how to calibrate their display.

Similarly, we can prevent an undesirable context from the technical perspective, for example for video QoE assessment, by pre-loading videos with included distortions in the remote browser, so that additional distortions introduced by the transmission do not affect the playback. And thus influence of the users' context with respect to bandwidth is no longer an issue.

3.5.2 Expectation of Users

A hidden influence factor on the user level can be the users' expectations: those used to lower quality (e.g. low video resolution) will rate differently than those typically consuming higher quality (e.g. high video resolution). The expectation level may be closely related to the country of the subject and users from different regions may have different expectations about the provided content quality. In general, we have two options to cope with expectations. We can either quantify the degree of expectations or we can reduce the expectations by instructing the test user accordingly. One option to quantify the expectations is to group users according to their expectations by asking them about their habits and typical use of a service, for example, "How often do you watch Internet videos?" and "Do you watch low or high resolution in YouTube?", respectively, where the assumption is that subjects who do not use video streaming services often may be more tolerant to worse quality.

In the QoE rating task, a user may additionally be asked to rate on an extra expectation category scale that is better aligned with the actual user's expectations. The subjects then rate the perceived quality with, for example, five levels of expectations: *(-2) Much worse than I expected. (-1) Worse than I expected. (0) Just as I expected. (1) Better than I expected. (2) Much better than I expected.* This rating scale is accompanied with a question regarding the perceived quality, e.g. *"Please indicate to which*

degree the overall quality of this video was in line or not in line with your expectations? The overall video quality was...". Still, the quantification of expectation remains a topic for future work.

3.5.3 Demographics and User Impairments

There are also several options for measuring demographic data that may have an impact on the QoE results and should therefore be statistically analysed.

3.1 Surveys, but the user may not give correct answers.

3.2 Extraction of data from social networks, but information is also not reliable.

3.3 Consistency tests to derive relevant information, but only a subset of data can be retrieved in order to avoid overusing consistency questions about demographics.

3.4 Get the information from crowdsourcing platform, if available.

Furthermore, hidden influence factors on the QoE results may be caused by physical impairments of the subjects if they are crucial for the study. For visual stimuli, for example, a test for colour blindness may be necessary to confirm normal colour vision.

3.5.4 Hard- and Software Environment

QoE crowdtesting are subjective tests conducted in a heterogeneous and therefore partly uncontrolled environment. Thus, monitoring on the system level is required to analyse hidden influence factors on a system level. Due to bottlenecks at the end user devices in terms of CPU, memory, or network bandwidth, additional artefacts may arise and affect the user rating accordingly. For example, the user's Internet access bandwidth may not be large enough to conduct a video quality test without stalling. However, those stalling events and the corresponding freezing of the video will impact the QoE. To overcome the impact of the network delay due to Internet delivery of data, the test application and data may be completely downloaded before the actual user test starts. Even so, the resulting initial delays may also be too long and influence the user rating. In both cases, it is evident that monitoring on system level is required. As a possible solution, download speed and latency may also be measured before the actual test, and then only users are selected with suitable connection speed and latency.

4 Implementation and Design of QoE Crowdtesting Campaigns

While designing a QoE crowdtesting campaign, the well established recommendations for laboratory subjective assessments can be respected only to a certain extent. Time constraints and test complexity should be adjusted in regards to a web based or other crowdtesting scenarios and to the variety of testing subjects among the crowd. Moreover, QoE crowdtesting brings additional requirements on server capabilities, computing power, and resource management. Hence we discuss major challenges concerning the available resources, either on a server side or on a client side, and best practices regarding the implementation (Section 4.1), as well as setting up the campaign (Section 4.2). A sophisticated two-stage crowdtesting design is proposed and recommended (Section 4.3).

4.1 Implementation

4.1.1 Test Server

The general approach for QoE crowdtesting is usually the use of a dedicated test server. This allows for a specific and well controlled testing environment. The choice of a dedicated test server gives additional possibilities to perform proper application layer monitoring, which further enhance the overall efficiency and accuracy of the given QoE crowdtesting application. Moreover, supporting technologies, for example, social networking or crowdsourcing platforms' APIs can be easily implemented.

Depending on the actual requirements on computing power and network resources, third-party services, such as cloud computing services or content delivery networks (CDN) can be utilised. The choice of a third party cloud service strongly depends on the size and type of a targeted QoE crowdtesting panel of users. While the use of a dedicated testing server is beneficial with respect to the better control of the environment, the test designer should take into consideration that users in a QoE crowdtesting campaign are accessing the application from a whole variety of different places. Hence, the accessibility of the server is an important issue. CDNs are well adapted to this fact and allow for better accessibility of the whole application. A large number of participants in a survey can result in a significant amount of a web traffic and this should also be taken into account when designing the recording system for the results, in particular with respect to the capability of handling a high number of queries in a short period of time.

Apart from cloud services or CDNs, another suitable option is limiting the number of simultaneous users of the application. If insufficient computational power or network resources are available, it is beneficial to put the users in a waiting queue and inform them about the waiting time. However, a long waiting time could result in a decrease of successfully finished surveys. Also, the options for using cloud services or waiting queues are not mutually exclusive and, if needed, they can be combined.

4.1.2 Client Interface

The client interface may change significantly in the QoE crowdtesting, depending on the users' environment. This should be reflected in the basic application design and the implemented technologies should put minimal requirements on browser's capabilities. In particular, the designer should focus on widely available and adopted technologies, since users may access the application from locations such as Internet cafes, where they are unable to install additional software.

According to [51], the technologies mainly supported in web browsers are still Flash and JavaScript, while Java is on a decline, with only approximately 70% of the market share. Similarly, other technologies like Silverlight, QuickTime or Mediaplayer are representing less than 50-70% of the market share. Thus if the application requires support of not commonly used technologies e.g Java, it will cause substantial loss of workers who successfully finish the assigned task. Losing 30% of the subjects in a QoE crowdtesting survey can easily represent hundreds of people, causing a bias in the overall results and

demographics of the crowd.

Support for all the required features should be properly tested before the beginning of the actual QoE assessment. These tests should include basic benchmarks, to ensure the browsers' ability to display the test correctly. For example, the video QoE crowdtesting application of Gardlo et al. in [52] included tests for JavaScript and Flash support in the browser, Internet connection speed, screen resolution and the flaw-less playback of high definition videos. Note that an important point is to keep the benchmarking time as short as possible, as not to interfere with the actual QoE crowdtesting, and thus distracting the users. Moreover, as already stated earlier, waiting time is one of the key influence factors on QoE which can introduce a bias to the overall QoE results.

4.2 Setting up the Campaign

In the process of creating and setting up the new campaign, a fundamental question is the length of the test. Despite the apparently rudimentary character of the question, the answer is rather complex, as the length of the test strongly correlates to several parameters, namely the

- overall *enjoyability* of the whole survey,
- complexity of the task,
- user interface,
- amount of reward,
- user's ability to understand the task.

Hence, the overall length of the survey is a combination of the parameters above. Generally, the more the user enjoys the task, the longer the test can last. This is strongly related to the key issue of incentives and payment schemes in Section 3.4.

Similarly to the selection of the required technologies, the designer should stick to a very basic and transparent design, with minimal requirements for user interactions, preferably many of them automatized e.g. automatic control of the input forms, measurements of connection speed as a background process or extraction of demographic data from the users' social network profile. Regardless of the depth of the integration of automatized interactions, it is necessary to keep in contact with the user and inform him about the task's progress.

It is beneficial to offer users the application interface in their native language, possibly even by only adding an automated translation plug-in to the page. Where not applicable, it is recommended to use simple English sentences. Most of the time, the user is not interested in the technical details of the application, and advanced terms can easily distract or confuse the workers. Real-time statistical analysis of the results can be included, so that the campaign can be stopped automatically as soon as a sufficient number of users finished the assigned task.

Workers should also be properly rewarded for the successfully finished task, with respect to the complexity and overall duration of the task. In [24], users get only paid depending on specific rules after successfully completing a task and achieving sufficiently high reliability scores. Better paid tasks will attract more users, but they will also be more critical to the application. Workers also tend to gather in virtual communities and share their experiences with certain employers, contributing to an employer's reputation and in extension the attractiveness of the tasks offered by this employer. Good payment and properly designed applications without any errors will be well received among the community, and this also helps to increase the overall efficiency of the QoE crowdtesting application. To support such communities, the designer is encouraged to implement a feedback channel e.g. via comments, a contact form or forums.

4.3 Recommendation: Two-Stage QoE Crowdtesting Design

Current platforms do not implement sufficient reliability mechanisms to ensure high data quality. Therefore, own test servers are required which allow to tailor the reliability mechanisms to the needs of the QoE assessment tests.

Even though some platforms allow for the automatic monitoring of user reliability using *gold* data, for QoE assessment *gold* data is not available in general, as the subjective assessment is done exactly because no ground truth is available. Yet, depending on the QoE assessment task, some secondary properties may be utilised as *gold*, e.g. in the evaluation of the influence of stalling in video streaming on QoE, the number of noticed stalling events can be used as *gold*, if the events were introduced artificially and it is known objectively that stalling occurred or not.

Additionally, a dedicated test server allows the application designer to implement social networks' APIs, for example, Facebook, and enables the employer to utilise the mutual advantages of each of these two distinctive crowdsourcing categories. We may, however, lose users without social network profile, but we also gain an important advantage by having demographic data available without any additional questionnaires. This reduces the overall testing time and the additional data can also be well used in assessing the users' reliability.

The general recommendation for campaign setting is the ***two stage*** design which is illustrated in Fig. 2. The ***first stage*** represents a very simple and easy to do task, which:

a) tests the reliability of the users,
b) gathers a huge panel of users,
c) gathers information about the users in the crowd,
d) has a duration of less than a minute and has low pay ($0.10),
e) can perform context monitoring: hardware or software, or perform user's training.

The intention of this stage is to create a *pseudo-reliable* group of users, who will be later invited to the actual crowdtesting task. An example of such an application is a simple screen quality test, where the user has to select visible pictures from a group of difficult-to-see or invisible images on a low quality screen. The task is easy to do, fast to finish and has low pay, so within a short period of time and with low costs it is possible

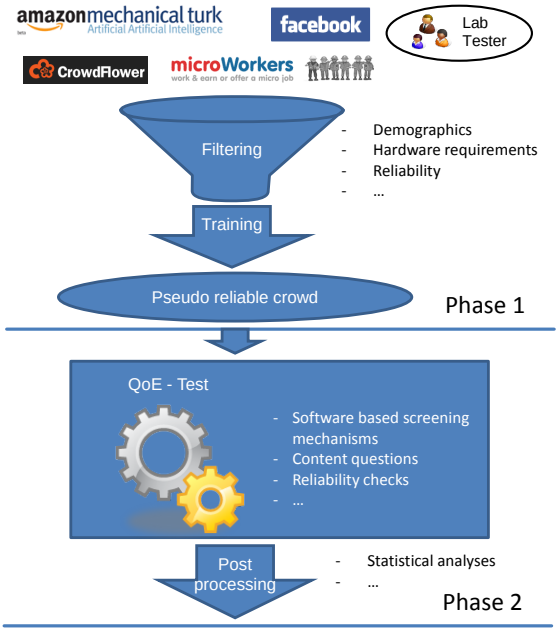


Figure 2: Recommended two-stage QoE crowdtesting design.

to create a reliable panel of users. This stage significantly improves the overall efficiency of the whole campaign. In our experiments, creating this pseudo-reliable panel increases the overall efficiency by more than 60 %.

The actual QoE test is then conducted in the ***second stage***, only with invited reliable users from a previous campaign. However, it is important to test if the same hardware or software is being used as in the first stage, but also to test the users reliability, for example, with content questions, demographic questions, or repetitive presentation of tested content. Note, that the use of hidden reference methods e.g. in ITU-R BS.1116 [53] can be considered as consistency questions as suggested in the two-stage design. This stage also requires higher reward for the workers. In the notion of ITU-R BS.116 [53], we also apply a pre-screening and a post-screening technique. The major argument for introducing the pre-screening, i.e. the first stage, is to reduce costs of the overall campaign and to get a pseudo-reliable crowd, while the post-screening in the second stage is required to ensure a reliable data set.

Although not necessary, it is sensible that the task required of the workers in the first stage is related to the task in second stage, for example, if the main task in second stage consists of a visual quality assessment, the first stage should also consist of a task including visual stimuli as in a screen quality test mentioned in the example above. Moreover, this can avoid any disappointment by the workers, resulting in decreased reputation of the employer, if the tasks in the two stages are very different. Also not every worker passing the first stage may be willing to participate in the second stage. In [54], for example, up to 75 % of the workers passing the first stage declined to participate in the second stage.

5 Statistical Analysis of Test Results

The two-stage design and the general implementation guidelines recommended in the previous section address the key issues discussed in Section 3.3 and lead to an overall better reliability of the QoE crowdtesting results. But even tough more reliable, the results may still contain a certain amount of unreliable ratings and/or workers. Similar to any laboratory-based QoE assessment test, we therefore need to perform a statistical analysis of the results in order to identify these unreliable results. Unfortunately, methods based on user ratings commonly employed in subjective QoE assessment are not suitable in the context of QoE crowdtesting and we demonstrate the shortcomings of these methods on the example of two QoE crowdtesting studies on video streaming. Before evaluating the commonly used screening methods from literature in Section 5.3, we briefly introduce the details of two QoE crowdsourcing studies in Section 5.1, followed by a demonstration of the severe concealed influence of unreliable ratings on QoE results in Section 5.2. Appropriate metrics to report the reliability of the results of QoE crowdtesting campaigns are then suggested in Section 5.4 and an indicator for hidden influence factors or unreliable ratings is proposed with the SOS hypothesis as discussed in Section 5.5.

5.1 Existing QoE Crowdtesting Data for Further Evaluation

For the statistical analysis, the results from two different subjective user studies on video streaming are revisited with respect to reliability.

5.1.1 YouTube Experiments

Firstly, YouTube video streaming is considered, where impairments in the network are perceived as stalling of the video playout by the user. If the available network bandwidth is lower than the video bit rate, video transmission becomes too slow, gradually emptying the playback buffer until underrun occurs. If rebuffering happens, the user notices interrupted video playback, commonly referred to as stalling.

In the QoE crowdtesting campaigns [20, 27], different reliability mechanisms were implemented as discussed in Section 3.3. In particular, unreliable users are determined based on content questions, consistency questions, and gold data [20]. After watching a video, the users were asked to answer simple content questions about the video clip. For example, "Which sport was shown in the clip? A) Tennis. B) Soccer. C) Skiing." The users were asked about their origin country in the beginning and about their origin continent at the end of the test to check their consistency. As gold data, we included videos without any stalling and additionally asked participants: "Did you notice any stops to the video you just watched?". If a user then noticed stops, we disregarded his ratings. We monitored browser events in order to measure the focus time, which is the time interval during which the browser focus is on the website belonging to the user test. In order to increase the number of valid results from crowdsourcing, we displayed a warning message if the worker did not watch more than 70 % of the video. The users could decide to watch the video again or to continue the test. When workers became

aware of this control mechanism, the percentage of completely watched videos doubled and almost three times more workers could be considered reliable than without the system warning. In particular, we consider a user and all his ratings to be unreliable, if one of those questions is not answered correctly or the video focus time was shorter than the video duration.

To have a realistic test scenario, the video experience in the test should mimic a visit of the real YouTube website. To this end, an instance of the YouTube Chromeless Player was embedded into dynamically generated web pages. With JavaScript commands the video stream can be paused, a feature we used to simulate stalling. In addition, the JavaScript API allows monitoring the player and the buffer status, i.e. to monitor stalling on application layer, by checking the current state of the player. In order to avoid additional stalling caused by the test users' Internet connection, the videos had to be downloaded completely to the browser cache before playing. This enables us to specify fixed unique stalling patterns which were evaluated by several users.

During the initial download of the videos, a personal data questionnaire was completed by the participant which also includes consistency question from above. The user then sequentially viewed six different YouTube video clips with a predefined stalling pattern. Typical YouTube videos of various content classes like news, sports, music clips, cartoons, etc. were used in the tests. Thereby, the video clips had different resolutions, motion patterns and video codec settings, but a fixed length of 30 s. After the streaming of the video, the user was asked to give his current personal satisfaction rating during the video streaming on an ordinal 5-point absolute category rating (ACR) scale. More details on the test setup can be found in [20, 13].

5.1.2 H.264/AVC Experiments

Secondly, the visual quality of H.264/AVC compressed CIF format videos is evaluated via the QualityCrowd framework without implementing any reliability mechanisms [21]. For this study, we used the *EPFL-PoliMi* data set by de Simone in [55, 56] that provides a set of H.264/AVC encoded video sequences distorted with different packet loss ratios. The data set provides both the distorted video and the subjective ratings gained in a ITU-R BT.500 compliant laboratory using a single stimulus (SS) method with continuous five point quality scale according to ITU-R BT.500. We reimplemented the same quality scale in the web interface shown to the workers. In doing so, we can compare the subjective results from the QoE crowdtesting with the results from the laboratory setup where the same subjective testing methodology was used as in the QoE crowdtesting. To avoid additional distortions caused by the transmission to the crowd workers, the videos from the data set were compressed losslessly with H.264/AVC and then pre-loaded in the workers' browser before playback. Hence, the unreliable crowdtesting results for H.264/AVC can be compared with reliable lab results.

Table 1: Details on the existing QoE crowdtesting data sets.

	YouTube [20]	H.264/AVC [21]
impairment	stalling: video interruptions and waiting times	video quality degradation through packet loss
rating scale	ordinal 5-point ACR scale	continuous 5-point ACR scale
test method	single stimulus	single stimulus
ITU Rec.	ITU-T P.910	ITU-R BT.500
#videos	21 typical YouTube videos	28 CIF videos
video duration	30 s	10 s
#subjects	722	59
reliability	gold standard, consistency and content questions, video focus time	none, but 2 laboratory tests with 17 and 23 subjects

5.2 Influence of Unreliable User Ratings on QoE Results

For the YouTube tests, the unreliable user ratings are determined based on content questions, consistency questions, gold data and video focus time as described in the previous section. The results for the QoE crowdtesting campaign are depicted in Fig. 3 as a cumulative distribution function (CDF) of the unreliable user ratings as well as the corresponding 95 % confidence interval. The unreliable user ratings F can be approximated by a discrete uniform distribution U with values from 1 to 5. The average user rating $\overline{F}$ and the expectation value $\overline{U}$ is 3.04 and 3.00, while the standard deviation is $\sigma_F = 1.45$ and $\sigma_U = 1.58$, respectively. A Pearson's χ^2 test is performed with the null hypothesis that the unreliable ratings are uniformly distributed. At the 5 % significance level, the null hypothesis cannot be rejected with a p-value of 0.39 to observe the given statistic with probability p.

Fig. 4 depicts the influence of unreliable user ratings on QoE results. In particular, the mean opinion scores (MOS) and the corresponding 95 % confidence intervals for YouTube experiments are plotted depending on the test conditions, representing the number of stalling events during the YouTube video playout in that case. The reliable user ratings from the YouTube experiments (237 in total) are therefore considered in presence of a ratio of α unreliable ratings in relation to the overall number of ratings. The unreliable user ratings are drawn from a uniform distribution between 1 and 5 according to Fig. 3. The three different curves for α in Fig. 4 illustrate the following observations: firstly, the obtained MOS values look all reasonable. The presentation of MOS values only does

not allow to identify the validity and the reliability of the results. The results, however, clearly show a severe impact of unreliable users on the observed MOS values $\widetilde{R}_x$ for a test condition x. The true MOS value $\overline{R}_x$ for $\alpha = 0$ is shifted towards the average unreliable user rating $\overline{F} = 3$, with $\widetilde{R}_x = (1 - \alpha)\overline{R}_x + \alpha\overline{F}$. Unfortunately, many subjective user studies only present MOS values without quantifying the reliability. Often confidence intervals are misused to quantify the reliability of the user ratings, but a 95 % confidence interval for a MOS value only shows that the mean rating including the unreliable user ratings lies within the confidence interval with a probability of 95 %. Secondly, confidence intervals I may therefore even decrease in the presence of unreliable user ratings due to the increased number N of ratings in total, as $I \sim 1/\sqrt{N}$.

As a consequence of these two observations, we conclude that it is evident that reliability has to be quantified for QoE crowdtesting by appropriate means and that unreliable user ratings have to be filtered out which will be discussed in the next section.

5.3 Comparison of User Rating based Screening Mechanisms (URS) and Additional Reliability Mechanisms (ARM)

In literature there exist two overall categories of screening mechanisms: firstly, filtering of users based on the actual user ratings and secondly, screening of users, independently of the ratings, but with additional reliability mechanism e.g. consistency tests [20, 57, 58]. For brevity, we will abbreviate in this contribution *user rating based screening mechanism* with *URS* and *additional reliability mechanisms* with *ARM*. The ARM approach leads to extra effort in the implementation and in the analysis, however, unreliable user can be clearly identified. For that reason, we use the results from the YouTube experiments which follow the ARM approach and implement several reliability mechanism from our proposed two-stage crowdtesting design. Thus, we know reliable and unreliable users for the YouTube tests.

Then, we apply different URS screening mechanisms from literature and compare their ability to identify unreliable users for the YouTube results. URS screening methods can be roughly separated into at least two classes [53]: one is based on inconsistencies compared with the mean result and relies on the ability of the subject to make correct identifications, the second class primarily eliminates subjects who cannot make the appropriate discriminations. Considering the variability of subjects' sensitivities to different artefacts [59], however, caution should be exercised in using URS [53]. As we will see later, URS screening mechanisms based on user ratings are not sufficient for QoE crowdtesting and thus ARM is necessary for unreliable user identification.

ITU-R BT.500 [22] proposes to screen subjects with the β_2 test. It counts, whether the scores of subjects lie in an interval around the mean of all ratings for the corresponding test condition. The length of the interval above and below the mean is m-times of the standard deviation of all ratings for this test condition. The kurtosis coefficient β_2 is then used to determine if the user scores are statistically normal or not. The factor m is chosen to be $m = 2$ (normal distribution) and $m = \sqrt{20}$ (no normal distribution), respectively. Based on this count, a user is rejected. The computation of the β_2 test in ITU-R BT.500 is given in Algorithm 1 in the appendix. More details can be found in [22].

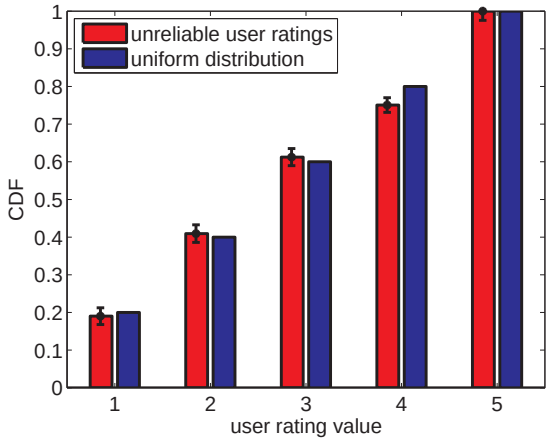


Figure 3: Unreliable user ratings for YouTube crowdsourcing tests identified by means of reliability mechanisms as described in Section 5.1 .

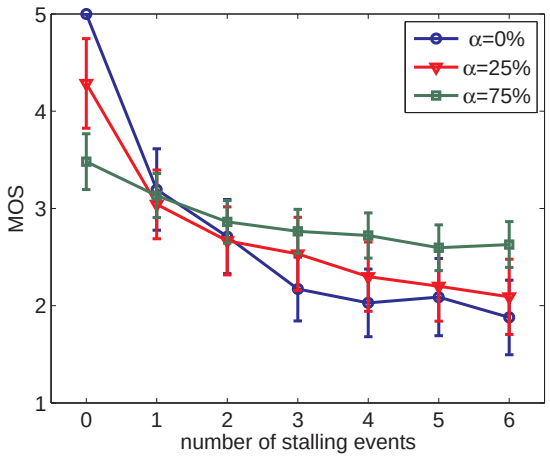


Figure 4: MOS values and corresponding 95 % confidence intervals for reliable YouTube ratings in presence of α unreliable ratings. The unreliable user ratings follow a discrete uniform distribution as in Figure 3 .

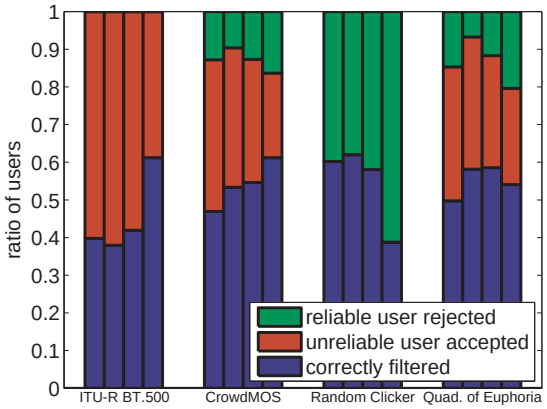


Figure 5: URS screening approaches filter out users based on their ratings. Application of those approaches to YouTube results shows that only a fraction of users is correctly identified. QoE crowdtesting requires ARM mechanisms.

Fig. 5 shows the results for four different YouTube crowdtesting campaigns. It can be seen that only half of the users are filtered correctly by the β_2 test. It accepts, however, also a large ratio of unreliable users. Hence, this method alone is not recommended for QoE crowdtesting.

In [60], the *crowdMOS* framework for subjective user studies proposes a URS screening mechanism: the sample correlation coefficient between the average user rating of a worker and the global average rating is used to identify unreliable users. The user ratings are averaged for the same test conditions e.g. number of stalling events in the YouTube experiments. A user is rejected, if the correlation coefficient is below a certain threshold, e.g. 0.25 in [60], see Algorithm 2 in the appendix. Then the global average rating is computed for the remaining users and the correlation coefficient is recomputed. Users are ranked in decreasing order of the correlation coefficient and the user screening starts again. Fig. 5 shows that the crowdMOS approach filters only half of the users correctly. A large fraction of unreliable users is accepted, which can be reduced by increasing the threshold. But an increased threshold would result in an even larger ratio of reliable users rejected.

Kim *et al.* in [61] investigate how to filter *random clickers* in a crowdsourcing-based study. In particular, Pearson's χ^2 test is applied to test the null hypothesis that the user is a random clicker, see Algorithm 3 in the appendix. The resulting p-value is used for excluding users having a high p-value above a certain threshold. In [61], a threshold of 0.02 is used. This approach seems quite promising, as the results in the previous Section 5.2 clearly reveal unreliable user ratings to be randomly clicked. Fig. 5, however, shows that the random clicker approach rejects many reliable user ratings, as they appear to be statistically random. This may be caused by the actual test design and the order of test conditions. Another explanation is the fact that users cannot differentiate the impact of the test conditions or perceive some test conditions equally.

Table 2: Comparison of URS screening methods based on user ratings.

Approach	Key measure for identifying reliable users
ITU-R BT.500 [22]	β_2 test counts the scores of a subject lying in an interval around the mean of all ratings for the corresponding test condition
CrowdMOS [60]	sample correlation coefficient between the ratings of a subject and the global average rating
Random Clicker [61]	p-value of Pearson's χ^2 test that the user ratings are random
Quadrant of Euphoria [24]	transitivity satisfaction rate by counting the triplets of user ratings u_A, u_B, u_C for conditions A, B, C following a transitive relation $u_A < u_B \wedge u_B < u_C : u_A < u_C$

While an increased threshold reduces the ratio of rejected reliable users, the increased threshold leads to a higher ratio of accepted unreliable users. We conclude that the analysis of random user ratings is not sufficient for unreliable user identification.

In [24], Chen *et al.* present the *Quadrant of Euphoria* which is a web-based crowdsourcing platform for QoE assessment. For filtering out subjects, a new metric is introduced, the *Transitivity Satisfaction Rate* (TSR). TSR is defined as the number of judgement triplets satisfying transitivity divided by the total number of triplets where transitivity may apply. Transitivity in this context means the reasonable assumption that preferences of users are a transitive relation. Hence, if a user prefers the test condition A to B and B to C, the user will normally prefer A to C and users are rejected, if the TSR value is below 0.8 [24]. The calculation of the TSR value is described as Algorithm 4 in the appendix. The obtained results are similar to the crowdMOS approach as a large ratio of unreliable users is accepted while some reliable users are rejected. Similarly to crowdMOS, the threshold value can be fine-tuned to reduce the acceptance of unreliable users, but at the cost of an increased rejection of reliable users.

We conclude that the URS approaches presented in this section and summarized in Table 2 cannot be used alone to clearly identify unreliable users. Hidden influence factors or the variability of subjects' sensitivities to different artefacts are not determined by those URS approaches. Although a combination of them may be interesting to improve screening, such as combining the random clicker approach and ITU-R BT.500, this remains a topic for future work. In summary, the screening of subjects should be done based on ARM methods as proposed in our two-stage design, which clearly identifies unreliable users independent of any hidden influence factor and the actual user rating. In [24], the payment of crowdsourcing users depends on a reliability metric in the form of the TSR above a certain threshold. Still, for the same reasons, it is recommended that payments are only refused when a subject is rejected according to the ARM methods.

Table 3: Statistical measures to quantify the reliability of the H.264/AVC test results.

Metric	EPFL	Polimi	Crowd
inter-rater reliability	0.928	0.905	0.699
Kendall's W	0.925	0.871	0.657
intra-class correlation	0.882	0.847	0.540
Krippendorff's α	0.874	0.837	0.484
average intra-rater correlation	0.922	0.956	0.818

5.4 Reliability Metrics for QoE Crowdtesting Campaigns

For any QoE crowdtesting study, appropriate metrics to report the reliability of the results of QoE crowdtesting campaigns are suggested. However, those reliability metrics do not necessarily allow to identify unreliable users, as they quantify the reliability of the entire data set from the QoE crowdtesting campaign. In order to quantify reliability, different well known metrics from statistics can be used. In particular, inter-rater and intra-rater reliability should be specified for any subjective user study, especially for QoE crowdtesting. *Inter-rater reliability* describes the degree of agreement among raters. For a QoE crowdtesting campaign, we define it as the absolute value of the Spearman rank-order correlation coefficient between all user ratings and the corresponding test conditions, as in [20]. Spearman rank-order correlation considers only that the items on the rating scale represent higher vs. lower values, but not necessarily of equal intervals. Hence, the inter-rater reliability returns a single value between 0 and 1 for a campaign.

Other well known metrics to quantify reliability are Kendall's W, the intra-class correlation coefficient (ICC), and Krippendorff's α. *Kendall's W* also known as Kendall's coefficient of concordance [62] is a non-parametric statistic ranging from 0 to 1. A value $W = 0$ indicates no agreement between the subjects, while $W = 1$ shows complete agreement. The *intra-class correlation* (ICC) [62] assesses the consistency of different subjective ratings for the same test conditions. Thereby, 0 indicates a complete lack of agreement and 1 indicates a perfect agreement between the subjects. *Krippendorff's* α [63] is a reliability coefficient to measure the agreement among test subjects. α ranges from 0 to 1. A value $\alpha = 0$ indicates the absence of reliability, while $\alpha = 1$ indicates that subjects agree perfectly.

For the H.264/AVC results, Table 3 shows the reliability metrics for the two laboratory studies conducted at Politecnico di Milano ('Polimi') and EPFL Lausanne ('EPFL') [55] as well as for the QoE crowdtesting study ('Crowd'). It can be seen that the laboratory studies lead to high values for the different reliability metrics close to 1, which means that all users are reliable. The observed reliability metrics for the QoE crowdtesting study, however, show significantly lower values. Low values of those reliability metrics may not be necessarily caused by unreliable users, but may also be an indicator for hidden influence factors. For example in [52], differences between crowdsourcing platforms have been reported which were caused by the heterogeneous environment: high definition videos were streamed to users who were asked to rate the visual quality, but small

screen resolutions of some users' displays did not allow for discrimination of the test conditions. Hence, only high reliability values imply that the test subjects are reliable and that no hidden influence factors exist.

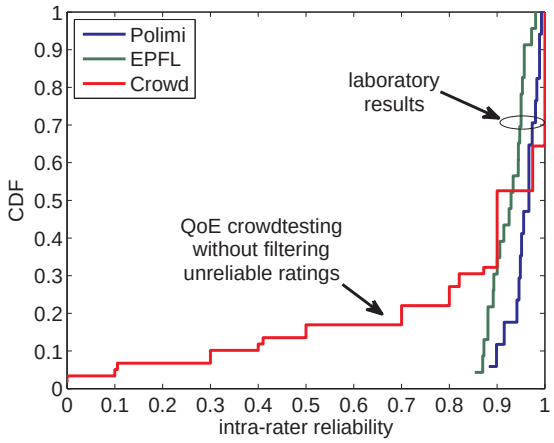


Figure 6: Intra-rater reliability of users for H.264/AVC lab and crowd studies.

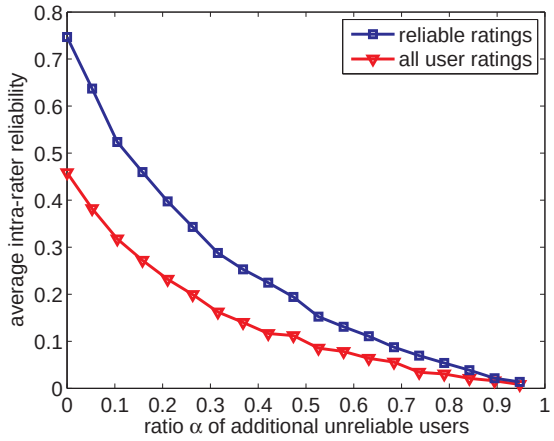


Figure 7: Average intra-rater reliability across all users and reliable users in presence of additional unreliable user ratings for YouTube QoE crowdtesting, respectively.

In contrast, *intra-rater reliability* determines to which extent the ratings of an individual user are consistent. Once again, the Spearman rank-order correlation coefficient is used for ordinal data between the user ratings and the test condition. Fig. 6 shows the CDF of the intra-rater reliability for the H.264/AVC crowdtesting studies as well as for two laboratory studies. It can be seen that the laboratory studies are close to 1, which means that all users are reliable. The QoE crowdtesting study, however, shows a large fraction of users with low values. This demonstrates the need for screening, which could be done on basis of this metric. But the definition of a user rejection threshold leads to similar difficulties as with the URS approaches in the previous section. Therefore, we

recommend ARM methods for screening.

Similar to [20], we define the *average intra-rater reliability* of a campaign by averaging the intra-rater reliability over all users. A high value also indicates that the user ratings are reliable and that no hidden influence factors exist. Fig. 7 shows the corresponding values for the YouTube crowdtesting studies across all users and the reliable users only, respectively. With α as the ratio of unreliable users with random ratings, we firstly consider the pure crowdtesting results, i.e. $\alpha = 0$. It can be seen that the value for all users is much lower than for the filtered users. This shows that the ARM methods applied in the crowdtesting campaigns are successful and that no hidden influence factors are contained in the study. Increasing the ratio α of unreliable users decreases significantly the average intra-rater reliability which converges towards 0 in the case of purely random ratings.

In summary, we conclude that reliability measures as proposed in this section should always be stated for filtered QoE crowdtesting studies. While high values show reliable user ratings, low values imply the presence of unreliable users or hidden influence factors in the QoE crowdtesting campaign.

5.5 Standard Deviation of Opinion Scores: SOS Hypothesis

Another approach to identify problems in a QoE crowdtesting campaign, i.e. unreliable users or hidden influence factors, is based on the SOS hypothesis [59] which considers the standard deviations of opinion scores (SOS). The SOS hypothesis postulates a square relationship between the variance $\mathrm{SOS}(x)^2$ and the corresponding MOS value x which depends only on a single factor, the so-called SOS parameter a. For a 5-point rating scale, the SOS hypothesis is described by the following equation.

$$\mathrm{SOS}(x)^2 = a(-x^2 + 6x - 5) \tag{1}$$

This SOS parameter has a typical value for different applications and impairment types [59]. For each test conditions we therefore obtain one particular MOS value and one SOS value over the corresponding user ratings for this test condition. For the entire campaign, we obtain one SOS parameter a. If the test condition contains a hidden influence factor or unreliable ratings, then the postulated SOS-MOS relationship cannot be observed. As a consequence, the unreliable users need to be filtered out and the hidden influence factors have to be found, respectively. In case of hidden influence factors, the test conditions have to be further differentiated according to those factors, then the SOS hypothesis applies again.

Fig. 8 shows the results for the H.264/AVC crowdtesting and laboratory study. It can be seen that the lab results clearly follow the SOS hypothesis. In case of the crowdtesting results however, no clear square relationship can be observed for the unfiltered data. This is due to the fact that unreliable user ratings are not filtered out and increase the SOS values due to the random rating. This can also be seen by the high values of a compared to typical values for various applications as given in [59].

As a result, the SOS hypothesis provides a simple mean to check for reliability problems in QoE crowdtesting.

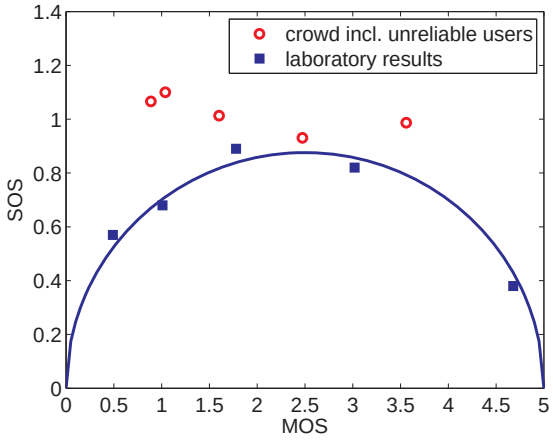


Figure 8: SOS hypothesis for H.264/AVC crowdtesting indicates unreliable users or hidden influence factors. Lab results follow the SOS hypothesis.

6 Best Practices for QoE Crowdtesting

Summarising the rules suggested in the last sections in the form of best practices for designing QoE crowdtesting campaigns, we can differentiate between three main categories:

- Technical implementation
- Campaign and task design
- Statistical analysis

The *Technical implementation* of the test should take into consideration the spread of the used technology among the targeted crowd. The use of widely available technologies, such as Flash player for video playback, are strongly recommended. Depending on required computational power, size of the crowd and/or geographical location, CDN networks and Cloud services can provide better service in comparison to a standalone server as discussed in Section 4.1.

To cope with the limited reliability of the crowd and other factors influencing the rating behaviour, for the *Campaign and Task design* we recommend the following steps:

1. The task should be designed to prevent cheating [37].
2. A pseudo-reliable crowd is created by simple, short, and cheap tests with different reliability elements. Only reliable users are then allowed to pass to the actual QoE tests with higher payments. This approach is also known as pilot task and main task [54].
3. Different elements need to be added in the task design to check the reliability of the users [20] and to filter out unreliable users in the first and second stage of the

QoE test. Combining these elements also leads to an improved reliability of the results [64]. Additional reliability mechanisms (ARM) include, but are not limited to:

3.1) Verification tests [57, 58], including captchas or computation of simple text equations: *"two plus 3=?"*, *"Which of these countries contains a major city called Cairo? (Brazil, Canada, Egypt, Japan)"*.
3.2) Consistency tests: First, the user is asked *"In which country do you live?"*. Later, the user is asked *"In which continent do you live?"*.
3.3) Content questions about the test: *"Which animal did you see?" (Lion, Bird, Rabbit, Fish)*.
3.4) Gold standard data [65]: *"Did you notice any stops to the video you just watched?" (Yes, No)*, when the actual test video did not include any stallings.
3.5) Application-layer monitoring [20]: Monitoring of response times of users and browser events to capture the focus time.

The important thing to keep in mind is not to add too many reliability items and questions, as otherwise the assessment task will become too lengthy. Further, too many of these questions may give a signal of distrust to the users. As a result, users may abort the survey. In general, incentives and proper payment schemes depending on the actual work effort are the key to high quality work. Incentive schemes such as gamification have the potential to make crowdsourcing an even more powerful tool to deliver high data quality in QoE assessment.

Regarding the best practices for evaluating the campaign and calculating the overall *statistics* of the crowdsourcing testing we encourage the use of a combination of URS and ARM mechanisms. URS methods alone cannot clearly identify unreliable users, since, for example, hidden influence factors or the variability of subjects' sensitivities to different artefacts are not determined as discussed in Section 5.3. We therefore recommend to use ARM approaches for screening of test subjects that are able to clearly identify unreliable users independent of any hidden influence factor and the actual user rating. Nevertheless, reliability measures such as inter- and intra-rater reliability or Krippendorff's α should always be stated for QoE crowdtesting studies, where high values show reliable user ratings, but low values imply the presence of unreliable users or hidden influence factors in the QoE crowdtesting campaign. We further recommend to include the SOS analysis of the results (Section 5.5) together with the information about the crowdsourcing platform used for the QoE assessment.

7 Conclusions and Outlook

We presented in this contribution best practices for QoE crowdtesting by providing a detailed discussion of the key issues faced in QoE crowdtesting and the corresponding remedies on the example of QoE assessment for video. In particular, we addressed the design, implementation and reliability assessment for successful QoE crowdtesting campaigns and compiled a set of best practices for crowdsourcing QoE testing.

Using these guidelines, subjective testing methodologies can be adapted to the crowdsourcing environment for QoE crowdtesting. Although the focus was on QoE assessment in this contribution, crowdtesting may also be used in other areas of interest, such as task performance or usability. The scope of these tests can contain any stimuli and is in general only limited by the capabilities of the crowd workers' devices and available bandwidth.

Open issues that need to be addressed in future work include methods to keep workers focused during longer tests, indicators of decreasing worker reliability in long duration tests, and lastly further studies of incentives to avoid cheating by the workers. Still, we believe that crowdtesting is an invaluable tool for any researcher working in the field of multimedia performing subjective testing, as it reduces significantly the resources necessary for validating new algorithms.

References

[1] E. Estellés-Arolas and F. González-Ladrón-de Guevara, "Towards an integrated crowdsourcing definition," *Journal of Information science*, vol. 38, no. 2, pp. 189–200, 2012.

[2] reCaptcha, July 2013.

[3] InnoCentive, July 2013.

[4] T. Hoßfeld, M. Hirth, and P. Tran-Gia, "Modeling of Crowdsourcing Platforms and Granularity of Work Organization in Future Internet," in *International Teletraffic Congress (ITC)*, (San Francisco, USA), Sept. 2011.

[5] Crowdflower, July 2013.

[6] CrowdSource, July 2013.

[7] Microtask, July 2013.

[8] TaskRabbit, July 2013.

[9] Amazon Mechanical Turk, Feb. 2013.

[10] Microworkers, Feb. 2013.

[11] Facebook, July 2013.

[12] P. L. Callet, S. Möller, and A. Perkis (eds), "Qualinet white paper on definitions of quality of experience (2012)," June 2012.

[13] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau, "Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience," in *Data Traffic Monitoring and Analysis: From measurement, classification and anomaly detection to Quality of experience* (M. M. Ernst Biersack, Christian Callegari, ed.), Springers Computer Communications and Networks series, 2013.

[14] T. Hoßfeld, R. Schatz, M. Varela, and C. Timmerer, "Challenges of qoe management for cloud applications," *Communications Magazine, IEEE*, vol. 50, no. 4, pp. 28–36, 2012.

[15] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J.-H. Hong, and A. Dey, "Factors influencing quality of experience of commonly used mobile applications," *Communications Magazine, IEEE*, vol. 50, no. 4, pp. 48–56, 2012.

[16] J.-S. Lee, F. De Simone, T. Ebrahimi, N. Ramzan, and E. Izquierdo, "Quality assessment of multidimensional video scalability," *Communications Magazine, IEEE*, vol. 50, no. 4, pp. 38–46, 2012.

[17] R. Schatz, T. Hoßfeld, L. Janowski, and S. Egger, "From Packets to People: Quality of Experience as New Measurement Challenge," in *Data Traffic Monitoring and Analysis: From measurement, classification and anomaly detection to Quality of experience* (M. M. Ernst Biersack, Christian Callegari, ed.), Springers Computer Communications and Networks series, 2013.

[18] "ITU-R BT.500 methodology for the subjective assessment of the quality for television pictures," Sept. 2009.

[19] "ITU-T P.910 Subjective video quality assessment methods for multimedia applications," Apr. 2008.

[20] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of youtube qoe via crowdsourcing," in *Multimedia (ISM), 2011 IEEE International Symposium on*, pp. 494 –499, dec. 2011.

[21] C. Keimel, J. Habigt, C. Horch, and K. Diepold, "QualityCrowd - A Framework for Crowd-based Quality Evaluation," in *Picture Coding Symposium (PCS) 2012*, (Krakow, Poland), pp. 245–248, May 2012.

[22] I. R. Assembly, "ITU-R BT.500-12 Methodology for the subjective assessment of the quality of television pictures ," 2009.

[23] C. Keimel, J. Habigt, C. Horch, and K. Diepold, "Video Quality Evaluation in the Cloud," in *Proceedings International Packet Video Workshop (PV) 2012*, (Munich, Germany), pp. 155 – 160, May 2012.

[24] K.-T. Chen, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "A crowdsourceable QoE evaluation framework for multimedia content," in *Proceedings of the 17th ACM international conference on Multimedia*, MM '09, pp. 491–500, ACM, 2009.

[25] K.-T. Chen, C.-J. Chang, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "Quadrant of euphoria: a crowdsourcing platform for QoE assessment," *Network, IEEE*, vol. 24, pp. 28 –35, Mar. 2010.

[26] C. Wu, K. Chen, Y. Chang, and C. Lei, "Crowdsourcing multimedia QoE evaluation: A trusted framework," *Multimedia, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2013.

[27] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea," in *QoMEX 2012*, (Yarra Valley, Australia), July 2012.

[28] Q. Xu, Q. Huang, T. Jiang, B. Yan, W. Lin, and Y. Yao, "HodgeRank on random graphs for subjective video quality assessment," *Multimedia, IEEE Transactions on*, vol. 14, pp. 844 –857, june 2012.

[29] C. Keimel, J. Habigt, and K. Diepold, "Challenges in crowd-based video quality assessment," in *Forth International Workshop on Quality of Multimedia Experience (QoMEX 2012)*, (Yarra Valey, Australia), July 2012.

[30] R. Schatz, S. Egger, and K. Masuch, "The impact of test duration on user fatigue and reliability of subjective quality ratings," *Journal of the Audio Engineering Society (JAES)*, 2012.

[31] T. Hoßfeld, R. Schatz, S. Biedermann, A. Platzer, S. Egger, and M. Fiedler, "The Memory Effect and Its Implications on Web QoE Modeling," in *23rd International Teletraffic Congress (ITC 2011)*, (San Francisco, USA), Sept. 2011.

[32] M. Hirth, T. Hoßfeld, and P. Tran-Gia, "Anatomy of a Crowdsourcing Platform - Using the Example of Microworkers.com," in *Workshop on Future Internet and Next Generation Networks (FINGNet)*, (Seoul, Korea), June 2011.

[33] S. Suri, D. Goldstein, and W. Mason, "Honesty in an online labor market," in *Human Computation: Papers from the 2011 AAAI Workshop (WS-11-11)*, 2011.

[34] J. Le, A. Edmonds, V. Hester, and L. Biewald, "Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution," in *SIGIR 2010 workshop on crowdsourcing for search evaluation*, pp. 21–26, 2010.

[35] D. Oleson, A. Sorokin, G. Laughlin, V. Hester, J. Le, and L. Biewald, "Programmatic gold: Targeted and scalable quality assurance in crowdsourcing," *Proc. HComp*, 2011.

[36] P. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *Proceedings of the ACM SIGKDD workshop on human computation*, pp. 64–67, ACM, 2010.

[37] A. Kittur, E. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pp. 453–456, ACM, 2008.

[38] L. Von Ahn and L. Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 319–326, ACM, 2004.

[39] C. Eickhoff and A. P. de Vries, "How crowdsourcable is your task," in *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 11–14, 2011.

[40] S. Dow, A. Kulkarni, B. Bunge, T. Nguyen, S. Klemmer, and B. Hartmann, "Shepherding the crowd: managing and providing feedback to crowd workers," in *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pp. 1669–1674, ACM, 2011.

[41] G. Little, L. Chilton, M. Goldman, and R. Miller, "Turkit: tools for iterative tasks on mechanical turk," in *Proceedings of the ACM SIGKDD workshop on human computation*, pp. 29–30, ACM, 2009.

[42] P. Dai, D. Weld, *et al.*, "Decision-theoretic control of crowd-sourced workflows," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[43] A. Kittur and R. Kraut, "Harnessing the wisdom of crowds in wikipedia: quality through coordination," in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pp. 37–46, ACM, 2008.

[44] M. Hirth, T. Hoßfeld, and P. Tran-Gia, "Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms," *Mathematical and Computer Modelling*, 2012.

[45] C. Eickhoff and A. P. de Vries, "Increasing cheat robustness of crowdsourcing tasks," *Information Retrieval*, pp. 1–17, 2012.

[46] M. Sabou, K. Bontcheva, and A. Scharl, "Crowdsourcing research opportunities: lessons from natural language processing," in *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*, p. 17, ACM, 2012.

[47] C. Harris, "Youre hired! an examination of crowdsourcing incentive models in human resource tasks," in *WSDM Workshop on Crowdsourcing for Search and Data Mining (CSDM)*, pp. 15–18, 2011.

[48] A. D. Shaw, J. J. Horton, and D. L. Chen, "Designing incentives for inexpert human raters," in *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, CSCW '11, (New York, NY, USA), pp. 275–284, ACM, 2011.

[49] L. Von Ahn, "Games with a purpose," *Computer*, vol. 39, no. 6, pp. 92–94, 2006.

[50] C. Eickhoff, C. G. Harris, A. P. de Vries, and P. Srinivasan, "Quality Through Flow And Immersion: Gamifying Crowdsourced Relevance Assessments," in *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval 2012*, ACM, Aug. 2012.

[51] S. Owl, "Java market share," Oct. 2012.

[52] B. Gardlo, M. Ries, T. Hoßfeld, and R. Schatz, "Microworkers vs. Facebook: The Impact of Crowdsourcing Platform Choice on Experimental Results," in *QoMEX 2012*, (Yarra Valley, Australia), July 2012.

[53] I. R. Assembly, "ITU-R BS.1116-1, Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems," 1997.

[54] M. Soleymani and M. Larson, "Crowdsourcing for affective annotation of video: Development of a viewer-reported boredom corpus," in *Proceedings of the ACM SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation (CSE 2010)*, pp. 4–8, July 2010.

[55] F. D. Simone, M. Tagliasacchi, M. Naccari, S. Tubaro, and S. Ebrahimi, "A h.264/avc video database for the evaluation of quality metrics," in *ICASSP*, pp. 2430–2433, IEEE, 2010.

[56] F. D. Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro, and T. Ebrahimi, "Subjective assessment of h.264/avc video sequences transmitted over a noisy channel," in *QoMEX 2009*, (San Diego, CA, USA), July 2009.

[57] O. Alonso, D. E. Rose, and B. Stewart, "Crowdsourcing for relevance evaluation," *SIGIR Forum*, vol. 42, pp. 9–15, Nov. 2008.

[58] J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor, "Are your participants gaming the system?: screening mechanical turk workers," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, (New York, NY, USA), pp. 2399–2402, ACM, 2010.

[59] T. Hoßfeld, R. Schatz, and S. Egger, "SOS: The MOS is not enough!," in *QoMEX 2011*, (Mechelen, Belgium), Sept. 2011.

[60] F. Ribeiro, D. Florencio, C. Zhang, and M. Seltzer, "Crowdmos: An approach for crowdsourcing mean opinion score studies," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 2416–2419, May 2011.

[61] S.-H. Kim, H. Yun, and J. S. Yi, "How to Filter out Random Clickers in a Crowdsourcing-Based Study? (Research Paper) ," in *BELIV 2012 Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, (eattle, WA, USA), Oct. 2012.

[62] K. L. Gwet, *Handbook of Inter-Rater Reliability: The Definitive Guide to Measuring the Extent of Agreement Among Multiple Raters.* Advanced Analytics Press, 2012.

[63] A. F. Hayes and K. Krippendorff, "Answering the call for a standard reliability measure for coding data," *Communication Methods and Measures*, vol. 1, no. 1, pp. 77–89, 2007.

[64] B. Gardlo, M. Ries, and T. Hoßfeld, "Impact of Screening Technique on Crowdsourcing QoE Assessments," in *22nd International Conference Radioelektronika 2012, Special Session on Quality in multimedia systems*, (Brno, Czech Republic), Apr. 2012.

[65] P.-Y. Hsueh, P. Melville, and V. Sindhwani, "Data quality from crowdsourcing: a study of annotation selection criteria," in *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, HLT '09, (Stroudsburg, PA, USA), pp. 27–35, Association for Computational Linguistics, 2009.

Appendix: Pseudocode for Algorithms Used in Section 5

Algorithm 1: β_2 test in ITU-R BT.500 [22].

input : N number of subjects
J number of test conditions
u_{ij} rating of subject i for test condition j
output: acceptance of user i as reliable user

1 initialize: $P_i \leftarrow 0; Q_i \leftarrow 0;$
2 **foreach** *test condition* j **do**
3 compute mean score $\bar{u}_j = \frac{1}{N}\sum_{i=1}^{N} u_{ij}$;
4 compute standard deviation $S_j = \sqrt{\sum_{i=1}^{N} \frac{(\bar{u}_j - u_{ij})^2}{N-1}}$;
5 compute kurtosis coefficient $\beta_{2j} = \frac{m_{4j}}{(m_{2j})^2}$ with $m_{dj} = \frac{1}{N}\sum_{i=1}^{N} (u_{ij} - \bar{u}_j)^d$;
6 **if** $2 \leq \beta_{2j} \leq 4$ **then** normal distribution
7 **if** $u_{ij} \geq \bar{u}_j + 2S_j$ **then** $P_i \leftarrow P_i + 1;$
8 **if** $u_{ij} \leq \bar{u}_j - 2S_j$ **then** $Q_i \leftarrow Q_i + 1;$
9 **else** non-normal distribution
10 **if** $u_{ij} \geq \bar{u}_j + \sqrt{(20)}S_j$ **then** $P_i \leftarrow P_i + 1;$
11 **if** $u_{ij} \leq \bar{u}_j - \sqrt{(20)}S_j$ **then** $Q_i \leftarrow Q_i + 1;$
12 **end**
13 **if** $\frac{P_i+Q_i}{J} > 0.05$ *and* $\left|\frac{P_i-Q_i}{P_i+Q_i}\right| < 0.3$ **then**
14 reject subject i to be reliable
15 **else**
16 accept subject i to be reliable
17 **end**
18 **end**
19 After eliminating the scores of those subjects being unreliable, the mean scores, standard deviation and kurtosis coefficient are to be recomputed in line 3–5.

Algorithm 2: CrowdMOS screening approach [60].

input : N number of subjects
$\mathbb{J}$ test conditions seen by subject i
u_{ij} rating of subject i for test condition j
output: acceptance of user i as reliable user

foreach *test condition* $j \in \mathbb{J}$ **do**
  compute mean score $\bar{u}_j = \frac{1}{N} \sum_{i=1}^{N} u_{ij}$;
end
compute sample correlation coefficient R between ratings u_{ij} from subject i and average rating $\bar{u}_j$: $c \leftarrow R(u_{ij}, \bar{u}_j)$
if $c < 0.25$ **then** reject subject i to be reliable ;
else accept subject i to be reliable ;

Algorithm 3: Random clicker test [61]. Apply Pearson's χ^2 test under the null-hypothesis that subject is a random clicker and the responses are uniformly distributed.

input : N ratings of subject i
n ratings are divided among n cells
(with $n = 5$ in case of an ordinal 5-point scale)
u_{ij} rating of subject i for test condition j
$\mathbb{U}_i$ set of ratings of subject i
output: acceptance of user i as reliable user

initialize: $E_i \leftarrow \frac{N}{n}$
for *k=1* **to** n **do** $O_k \leftarrow 0$;initialize observed frequencies
foreach *user rating* $u_{ij} \in \mathbb{U}_i$ **do** $O_{u_{ij}} \leftarrow O_{u_{ij}} + 1$;
for *k=1* **to** n **do** $O_k \leftarrow \frac{O_k}{N}$;
$\chi^2 \leftarrow \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$
Compute p-value by comparing the value χ^2 to a chi-squared distribution with $n - 1$ degrees of freedom
if $p < 0.02$ **then** accept subject i to be reliable ;
else reject subject i to be reliable ;

Algorithm 4: Transitivity Satisfaction Rate (TSR) [24].

input : $\mathbb{U}_i$ set of ratings of subject i
$\mathbb{J}$ test conditions seen by subject i
u_{ij} rating of subject i for test condition j
output: acceptance of user i as reliable user

initialize: $nTest \leftarrow 0; nPass \leftarrow 0;$
forall *triplets* $(u_A, u_B, u_C) \in \mathbb{U}_i^3 \wedge A \neq B \neq C \in \mathbb{J}$ **do**
    count total triplets: $nTest \leftarrow nTest + 1;$
    if $u_A < u_B \wedge u_B < u_C \wedge u_A < u_C$ **then** triplet is satisfying transitivity
        $nPass \leftarrow nPass + 1$
    end
end
if $\frac{nPass}{nTest} > 0.8$ **then** accept subject i to be reliable ;
else reject subject i to be reliable ;

Appendix: Comparison of Crowdsourcing Platforms

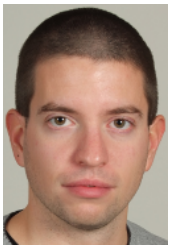
Feature \ Platform	MTurk	Microworkers	Facebook
Platform access	Only US residence are allowed to create campaigns	Support of international employers	Support of international employers
Diversity of participants	Mainly US and Indian workers	International users with a large portion from Asia	Mainly friends or acquaintances
Costs per task	One cent to a few dollars (depending of the task length)	Ten cents to a few dollars (depending on the task length)	free
Variable payment features	Bank transfer, Amazon.com gift cards	Micropayment services, wire card, credit card	Not applicable
Costs for qualification tests	free	Ten cents to a few dollars (depending on the task length)	Free
Effort to acquire a large amount of testers	None	None	Test has to be designed in a joyful manner to attract participants and to go viral
Time to acquire a few hundred of testes	Few hours to a few days	Few hours to a few days	Few days to a few weeks
Support of specialized participant groups	Worker groups can be selected by qualifications e.g. obtained by qualification test or overall performance, or by given attributes e.g. country	Worker groups can be selected by overall performance, special attributes e.g. country, and deliberately formed by selecting individual workers	No direct support of grouping participants
Integration of test into the platform	Forms are directly supported, more complex tasks have to be implemented on an own server and embedded in an IFrame	Only plain text descriptions and input is supported, more complex task have to be implemented on an own server	Tasks have to be implemented on an own server and can be embedded using an IFrame

Appendix: Biographies of the Authors

Tobias Hoßfeld is heading the FIA research group 'Future Internet Applications & Overlays' at the Chair of Communication Networks in Würzburg. He finished his PhD in 2009 and his professorial thesis (habilitation) 'Modeling and Analysis of Internet Applications and Services' in 2013. He has been visiting senior researcher at FTW in Vienna with a focus on Quality of Experience research. His main research interests cover social networks, crowdsourcing platforms, content distribution networks and clouds, as well as investigations on Quality of Experience for Internet applications like Skype, YouTube, Web Browsing or cloud applications in general. He has published more than 100 research papers major conferences and journals, receiving 4 best conference paper awards, 3 awards for his PhD thesis and the Fred W. Ellersick Prize 2013 (IEEE Communications Society).

Christian Keimel received the B.Sc. and Dipl.-Ing. degree in electrical engineering and information technology from the Technische Universität München (TUM), Munich, Germany, in 2005 and 2007, respectively, and is currently pursuing a Ph.D. degree at the Institute for Data Processing at TUM. He is a Research Assistant with the Institute for Data Processing at TUM, and his research interests include crowdsourcing, Quality of Experience (QoE) assessment with a particular focus on video, and the application of multi-way data analysis methods in the context of QoE prediction models. Christian Keimel is a deputy work group leader in the European network on QoE in multimedia systems and services (QUALINET), where he is focusing on future application areas and use cases for QoE. In 2011 he received the best student paper award at the third international workshop on Quality of Multimedia Experience (QoMEX).

Matthias Hirth is research assistant in the FIA research group 'Future Internet Applications & Overlays' at the Chair of Communication Networks in Würzburg. He received his Diploma in Computer Science in 2009 from the University of Würzburg and is now working towards his PhD. His research interests cover social networks, with a focus on structural analysis and applications, and crowdsourcing. Hereby, the main focus lies on the development of new mechanisms for crowdsourcing platforms as well as finding new crowdsourcing use cases.



Bruno Gardlo is a research assistant at FTW Wien. During his master's studies he spent several months at the Vienna University of Technology, where he was dealing with audiovisual quality in mobile environment. In 2012 he finished his PhD. at the University of Zilina, focusing his research on crowdsourcing, audiovisual QoE and web-development. During his PhD studies, he spent several months at Queen Mary, University of London and at University of Würzburg. His current research interests cover user experience in emerging web applications, user assessments exploiting modern social networks and user tests designs performed via crowdsourcing platforms.

Julian Habigt is a research assistant at the Technische Universität München (TUM), Institute for Data Processing. He received his Diploma in Electrical Engineering in 2009 from TUM and is now working towards his PhD at the Institute for Data Processing. His current research interests cover image processing, signal processing, information fusion and state estimation with a particular focus on view synthesis.

Klaus Diepold was born, raised and educated in Munich, Germany. He received a Dipl.-Ing. degree and a Dr.-Ing. degree both in Electrical Engineering from Technische Universität München (TUM) in 1987 and 1992, respectively. In the years 1993-2002 he worked in the video signal processing, television and video compression industry as technical director and entrepreneur in Munich, Oslo and New York. In 2002, he joined TUM as a full professor for data processing in the Department of Electrical Engineering and Information Technology. His research interests are in multimedia signal processing, data analysis for quality of experience and machine learning algorithms for cognitive systems. He is the Scientific Director of the Center for Digital Technology and Management (CDTM).

Phuoc Tran-Gia is professor and director of the Chair of Communication Networks, University of Würzburg, Germany. He is also Member of the Advisory Board of Infosim (Germany) specialized in IP network management products and services. Prof. Tran-Gia is also cofounder and board member of Weblabcenter Inc. (Dallas, Texas), specialized in Crowdsourcing technologies. Previously he was at academia in Stuttgart, Siegen (Germany) as well as at industries at Alcatel (SEL) and IBM Zurich Research Laboratory.His research activities focus on performance analysis of the following major topics: Future Internet & Smartphone Applications; QoE Modeling & Resource Management; Software Defined Networking & Cloud Networks; Network Dynamics & Control.