Multimedia Tools and Applications manuscript No.
(will be inserted by the editor)

Platform for real-time subjective assessment of interactive multimedia applications

Bert Vankeirsbilck · Dieter Verslype · Nicolas Staelens · Pieter Simoens · Chris Develder · Piet Demeester · Filip De Turck · Bart Dhoedt

Received: date / Accepted: date

Abstract With the advent of cloud computing and remote execution of interactive applications, there is a need for evaluating the Quality of Experience (QoE) and the influence on this QoE of network condition variations, media encoding parameter settings and related optimization algorithms. However, current QoE assessment focuses mainly on audiovisual quality in non-interactive applications, such as video-on-demand services. On the other hand, where experiments aim to quantify interactive quality, the focus is typically targeted at games, using an ad-hoc test setup to assess the impact of network variations on the playing experience. In this paper, we present a novel platform enabling the assessment of a broad range of interactive applications (e.g., thin client remote desktop systems, remotely rendered engineering applications, games). Dynamic reconfiguration of media encoding and decoding is built into the system, to allow dynamic adaptation of the media encoding to the network conditions and the application characteristics. Evaluating the influence of these automatic adaptations is a key asset of our approach. A range of possible use cases is discussed, as well as a performance study of our implementation, showing that the platform we built is capable of highly controllable subjective user assessment. Furthermore, we present results obtained by applying the platform for a subjective evaluation of an interactive multimedia application. Specifically, the influence of visual quality and frame rate on interactive QoE has been assessed for a remotely executed race game.

Bert Vankeirsbilck is funded by a Ph.D. grant of the IWT – Vlaanderen. Chris Develder is supported in part by a post-doctoral fellowship of the Research Foundation – Flanders (FWO-Vl.).

B. Vankeirsbilck, D. Verslype, N. Staelens, P. Simoens, C. Develder, P. Demeester, F. De Turck and B. Dhoedt
Ghent University, Department of Information Technology (INTEC), Internet Based Communication Networks and Services (IBCN) – IBBT
Gaston Crommenlaan 8, bus 201, 9050 Ghent, Belgium
Tel.: +32 9 331 49 38
Fax: +32 9 331 48 99
E-mail: bert.vankeirsbilck@intec.ugent.be

P. Simoens
Ghent University College - Department INWE, Valentyn Vaerwyckweg 1, 9000 Ghent, Belgium

Keywords Interactive Media Quality Assessment · Quality of Experience · Interactivity · Subjective Quality · Thin Client Computing

1 Introduction

To assess the quality of multimedia applications, focus is currently put onto the perceived audiovisual quality and the influence of network circumstances and media encoder settings on this quality. In such assessments, typically representative video fragments are selected and encoded in different formats, according to frequently occurring parameter combinations (often leading to a large set of benchmark fragments). These encoded fragments are then presented in a well chosen order to the subjects (i.e., users that are judiciously selected or deliberately excluded from a population), who rate the sequences according to the simulated circumstance.

This method has proven to be very valuable for subjective assessment of audiovisual applications, and facilitates repeating the same experiment under identical conditions for different subjects. However, the approach can obviously not be adopted in the case of interactive applications, as the system behavior is heavily depending on the interaction, preventing to play back recorded content.

In this paper, a system for subjective evaluation of *interactive multimedia-oriented applications* is proposed. Using this system, applications are assessed in real-time, i.e., the subjects interact with an executing application while the parameters under study can be steered on-the-fly. This way, the subjects can evaluate the influence of parameter settings while actually performing the task. To accomplish this, the platform features real-time multimedia codec reconfiguration and network impairment provisioning that are remotely controllable. Runtime resource monitoring and recording of the user input and the graphics in the session enable post-experiment analysis.

The proposed platform is based on a thin client approach, where application logic is typically hosted on a shared high performance server, limiting the functionality of the client to forwarding user input over the network and receiving graphical results for presentation. The main contribution of this paper consists of proposing a generic approach to assessing real-time, interactive applications, whilst allowing runtime adaptations and (re)configurations to optimize QoE dynamically. The platform can be used complementary to existing subjective assessment methodologies, as it includes the assessment of interactivity experience, audio or video parameter settings, network circumstances, performance of adaptation algorithms aimed at coping with network degradations, and the ease of performing tasks, with real-time adaptation possibilities. Examples of experiment scenarios that are supported by the proposed platform include:

1. *Automated gameplay assessment:* A user plays a game under a given configuration of video parameters, e.g., frame rate and quantization parameter (QP). At regular time intervals, a questionnaire pops up requesting the user to score the gameplay in terms of, for example, playing experience. When the question is answered, new parameter settings are configured and the experiment continues. The results of the experiment are used to design a second round of experiments in which algorithms are evaluated that mitigate network variations using the video parameters.

2. *Administered subjective interactivity experiment with network impairment:* The conductor of the experiment is seated next to the test subject and instructs him to perform tasks with the interactive application. Gradually degrading from perfect media encoding quality, the test subject is requested to mark the threshold of usability, i.e., for which the impairments make the interaction quality no longer acceptable. The conductor of the test is interested in creating a metric that links visual information and user input to the perceived interactive quality. Therefore, during the session, the experiment conductor controls both network and coding settings, and the screen is captured and the user input is recorded for possible correlation afterwards.
3. *Test subject finds subjective optimal parameter constellation given various impairment settings:* In this case, the test scenario is unsupervised, and a subject is given the task to interact with a given application. One parameter is constrained, e.g., the network bandwidth is configured to a specific value and the test subject needs to alter, in real-time, other parameters to a joint optimum, e.g., lowering frame rate to be able to increase per-frame quality or vice versa. Therefor, a specific interface is provided to fine-tune the parameters, such as a slider element in a controller program that allows the subject to adjust the balance between frame rate and per-frame quality.
4. *Collaborative applications:* Multiple test subjects are requested to interact with a collaborative application, such as a text editor that allows simultaneous input from multiple users. The parameters for all users can be set identical or different, depending on the experiment context. At regular time intervals, the users are requested to rate the interaction with the application, after which new settings are loaded for a new experiment iteration.
5. *Defining novel objective quality metrics:* In order to define novel objective quality metrics for interactive media, a researcher is interested in finding new correlations between interactive experience of a previously investigated use case and parameters that were not available or were not taken into account before by the researcher. The necessary environment measurement probes and parameter controls are, if not yet present, added to the platform, and activated. Using these additional interactivity measurement and control parameters, the previously executed experiments are easily repeated and extended due to the ability to retake the parameter configurations used. Hence, the results can be related to the previous findings to derive new quality metrics.

The *"Automated gameplay assessment"* scenario was selected as use case to study using the platform. The approach and results of this study are presented in Section 6.

The remainder of this paper is structured as follows. Section 2 presents related work in the field of subjective assessment of interactive multimedia applications. In Section 3, requirements for the assessment platform are defined and the architecture of our proposed system is presented. Also, example use cases and configurations of the platform are discussed. In Section 4, the details of our implementation are presented, followed by an experimental performance analysis in Section 5. Section 6 presents an example method of assessing an interactive media application using the proposed platform. Finally, Section 7 concludes our contribution, while identifying opportunities for future work.

2 Related Work

The International Telecommunications Union (ITU) provides recommendations for subjective assessment methods [9,10]. However, the focus of these recommendations is to provide subjective assessment methods for unidirectional audio and video. The interactive assessment recommendations on the other hand, are currently limited to conversational applications, such as audio and video phone calls between people, not addressing other types of interactive applications where responsiveness is crucial.

The OneClick framework presented in [3] enables capturing users' perceptions when they are using network applications. The main innovation of the framework is the acquisition of a specific discomfort metric, i.e., letting the subject perform a dedicated action such as a button press when he feels discomforted in the interaction with the application. Although real-time usage of the application is not excluded by the framework, the focus is clearly on this metric, the synchronization between user feedback times and the actual perception times of discomfort, and the analysis of the measurements. The test setup and real-time parameter configuration are implemented for this particular use case. Other subjective assessment platforms focus mainly on simplifying the recruitment of test subjects by eliminating the need to physically travel to the test lab [4,22]. Besides losing control over the test environment, these platforms concentrate on non-interactive applications.

A survey of visual quality assessment algorithms and an outlook for future research are given in [18], indicating advanced metrics and different methodologies used for assessing video quality. However, the applicability of these metrics and methodologies for interactive multimedia applications is not evaluated. In [16], an overview of techniques to measure QoE is presented, indicating that games are the only type of interactive application investigated in literature. In contrast, the platform presented in the current paper allows assessment of a broad range of interactive applications, not restricted to games, but includes image or video editing, text editing, spreadsheet data manipulation, web browsing and more. The platform enables to investigate the suitability of existing QoE metrics and to design new metrics to assess interactivity QoE.

The thin client principle, that separates the actual execution logic of the application from the presentation using network infrastructure, is frequently applied for remote execution of applications. In this field, studies have been performed to quantify the interactive quality obtained using such systems. A common approach in literature is to record the user's interactive session and replay this session multiple times under different system configurations [19,24,29,31]. The interactive response time is evaluated by comparing the times at which similar screen updates occur in each of the replayed sessions. In all of these studies, the interactivity is mainly based on a measurement of end-to-end latency, i.e., the time between generation of user input and the presentation of the related graphical update, giving an indication of responsiveness to user input. In [12], the authors argue that profiling the execution time spent in each method of a program and analyzing the end-to-end latency is insufficient for tuning interactive applications and focus on deeper, more directed profiling of latencies. Still, latency is not the only metric that constitutes interactive QoE. In our opinion, other metrics should be included in the interactive QoE assessment, such as visual quality and the characteristics of the performed task.

In [1], the suitability of thin client protocols for games is assessed in terms of achieved frame rate as a function of network delay, packet loss rate and available bandwidth. In view of the current explosion in cloud-based gaming, a survey of the degradation of the performance due to network latency and packet loss on QoE is presented in [11]. In the field of gaming experience in general, many studies have assessed the effect of network impairments on game experience [7,20,21,28]. In most cases, an indirect application level metric is used to quantify this experience, e.g., the kill/death ratio for a shooter game or lap times in a racing game. This approach is not applicable to other application types, especially when the concept "score" is not well-defined. Alternatives for measuring the interactivity QoE that are not restricted to games consist of logging usage times (departure rate) [2] or mean task execution times [30]. As argued earlier, the focus of these studies is primarily on the impact of network conditions, and for all of them, a custom experiment setup has been built.

Although most gaming experience literature emphasizes on first person shooter games, in [6] the authors show that the perspective of the shooter game has an influence on the perceived QoE. For instance, the difference is made between shooter games with a first person, third person and aerial perspective, as well as omnipresent games. The results also indicate that during different phases in the games, the requirements of the game on the network, and the expectations of the users differ. In [5], this idea is refined by investigating how the graphical characteristics differ in terms of visual motion and scene complexity for the gaming application domain specifically, depending on the gaming perspective (e.g., third or first person). They show that using these characteristics, it is possible to obtain perspective classification of shooter games. Together with the research performed in the field of Context of Use (e.g., [14]), these results re-enforce our idea that the perceived content delivery quality will differ depending on the characteristics of the application, and even further within one application, on the specific characteristics of the performed task.

As a conclusion, we can state that the assessment of interactive media is mostly restricted to games, and that these assessments are conducted using dedicated set-ups and platforms. The platform proposed and evaluated in the current paper however allows to assess the QoE achieved with various types of interactive applications, in presence of varying runtime conditions and accompanying adaptations from mitigating algorithms.

3 Platform Architecture

3.1 Requirements

We identified following requirements for a platform addressing subjective quality assessment of interactive multimedia applications, by analyzing the shortcomings of existing QoE frameworks:

1. *Configuration of network conditions:* parameters such as bandwidth limitations, jitter, packet loss or latency must be controllable to assess their impact on interactive media quality.

2. *On-the-fly adaptation of media encoder parameters:* for subjective study of the impact of adaptation algorithms.
3. *Modular design:* to supply adequate extensibility, e.g., towards different media codecs and adaptation algorithms.
4. *Real-time user interaction:* as the interactivity is the targeted focus of studies using the platform.
5. *Experiment replay:* in order to derive meaningful statistics from the test results, provisions must be made to repeat the same test for different users.

3.2 Architecture details

Based on the requirements highlighted above, we have designed a platform that is inspired by the thin client computing paradigm. However, conventional thin client systems are optimized for office-type scenarios and are not well-suited to display multimedia content [17]. Therefore, the audiovisual encoding functionality was implemented such that more appropriate media encoders can be supported. On the other hand, the handling of user input and dispatching these events to the applications as well as the provisions for intercepting the application output was retained. To have absolute control over network parameters, the client and server components can be deployed on the same node, and the network is emulated in terms of bandwidth, jitter, packet loss and latency.

The different functional blocks that comprise the architecture, and the relations between them, are presented in Fig. 1. Using the conventional thin client terminology, the entity that interacts with the user, i.e., that captures user input and presents audiovisual output, is called the *viewer*. The encoded user input is sent over the network to the entity that contains the application logic, called the *server*. The server hosts different applications concurrently, receives user input and dispatches the associated events to the correct application. These applications operate in the X-Server environment that handles X Drawing commands to write the output into the frame buffer. Specific support must be provided for Graphical Processing Unit (GPU) hardware accelerated applications (such as Application X in the diagram), as these normally do not write their output to the frame buffer. Hence, a dedicated component reads back this application output from the graphics card to write the output to the frame buffer, from which the media encoders take their input samples by polling the content. Media encoders are executed in a separate thread to avoid delay and interference of the other components, and to achieve independent control over, e.g., frame rate for video encoders. The encoded media network stream is pushed over the network for decoding at the viewer side. The control commands and encoded media stream are transported over separate channels, and can thus use different transport protocols and possibly distinct interfaces. In the server, a separate control loop is spawned, that waits for control commands on a separate network socket. Using this control loop, test parameters related to media encoder settings and network impairment can be altered. The protocol for this messaging is not a priori fixed: any connection based technology is possible, preferably a reliable one to ensure deterministic operation of the system.

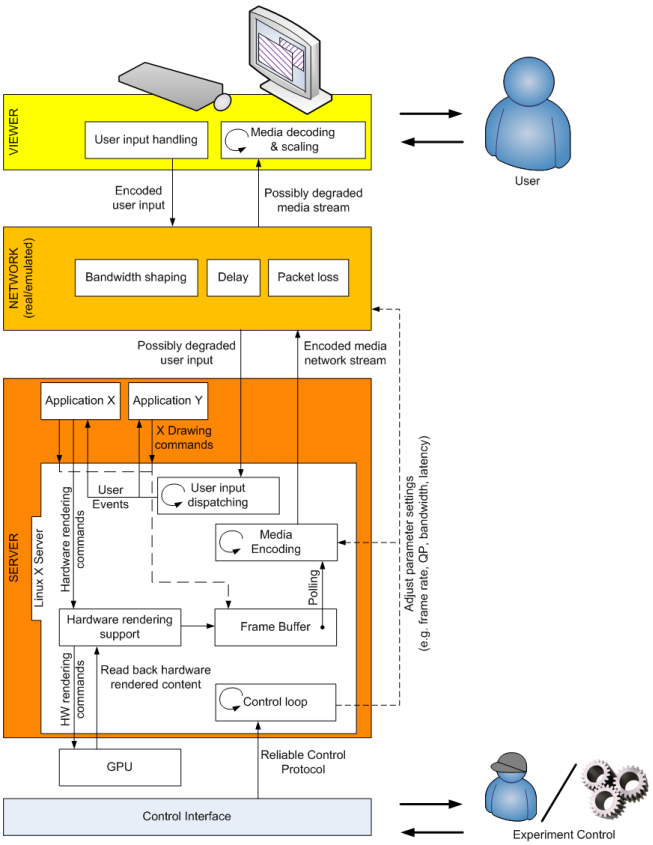


Fig. 1 Components of the subjective evaluation platform and their interactions.

3.3 Platform configuration options

Different test purposes imply different parameters that need to be controlled. In this perspective, next to supplying an extensible control protocol, the architecture allows to model the network in different ways. The network influence can be eliminated completely, by co-locating the viewer and the server on the same physical machine. To study the impact of network limitations (e.g., in terms of bandwidth or delay), a network emulator is plugged in between the viewer and the server. For validation purposes, a real network can be used, at the cost of losing exact reproducibility of the tests due to the uncontrolled environment. Furthermore, multiple viewers can be connected to one server in a star topology, e.g., for collaborative application testing. When identical parameter settings are allowed, broadcasting the graphical responses can be performed and synchronization between the questionnaire applications should be included to correctly set the parameters. When different parameter settings are required per user, separate media encoders need to be configured per user, hence simple broadcasting of the encoded graphics is insufficient, but each user has an individually controlled media codec, eliminating the need for synchronization.

Table 1 Technologies incorporated in our implementation

H.264 codec	x264 0.120.x *ultrafast* profile, *zero latency* tuning
ffmpeg software scaling	ffmpeg version 0.9.1.git-1869237
OpenGL support	VirtualGL 2.3
Control protocol	cJSON library

4 Implementation Details

A reference implementation of the architecture was implemented using the technologies listed in Table 1. Currently, the x264 implementation of the H.264 video codec has been integrated into the platform. Conversion between video formats for transmission and presentation on screen was handled by *swscale*, which is part of the *ffmpeg* package. Support for OpenGL based applications is provided by VirtualGL [27], that enables reading back the graphics rendered by the Graphics Processing Unit (GPU) into the frame buffer [8, 23]. We have implemented a control channel that uses JavaScript Object Notation (JSON) [13].

The implemented platform provides the following features:

1. *Real-time media codec reconfiguration:* encoding parameters, such as frame rate or quantization parameter for a video encoder, can be set dynamically. If the codec does not support such adaptations at runtime, this can be achieved by stopping and starting it on-the-fly.
2. *Codec support:* different real-time codecs can be plugged in easily. RGB frames are originally available in the server, which are currently converted to the YUV format using ffmpegs swscale library, for encoding with x264.
3. *Network impairment provisioning:* the platform supports any network emulator, such as a click modular router [15] or the Linux traffic control package (*tc* and *tcng*). This emulator can introduce packet loss, bandwidth drops or latency for example.
4. *Real-time monitoring:* is realized with conventional Linux tools, such as *tcpdump* or *wireshark* for network monitoring and *top* or reading *cpustat* for CPU measurements. Also, the viewer and server are instrumented to record the achieved frame rate and the end-to-end latency to estimate the interactivity objectively.
5. *Remote control:* the reconfiguration of the system can be controlled remotely. This remote control can also be embedded in the questionnaires, such that after each evaluation of a particular test case, the settings can be changed on-the-fly when starting the subsequent evaluation round. If necessary or appropriate, the screen can be divided in independent or synchronized video encoded regions, each with their own settings, e.g., for multiple-stimulus methodologies.
6. *Hardware accelerated application support:* the use of VirtualGL facilitates using hardware accelerated applications so that any application can be subjectively and interactively tested.
7. *Session recording:* to support offline comparison of the test results to existing subjective and objective video quality metrics, recording both server and viewer graphical information is possible, as well as the key strokes and mouse events of the user.

8. *Subjective assessment method support:* all frequently used subjective assessment methods are supported, and there is potential for expanding the set due to the flexibility of the platform. For instance, it is possible to set up a paired comparison experiment by starting two independent codecs in one session, that transfer the graphics with different configurations to separate screens, or could even be configured to encode half of the session's graphical output each for presentation on one screen. Approaches similar to the one-click framework presented in [3], where the users express discomfort while using the application in combination with a staircase design, are supported by the real-time controllability of the platform.
9. *Extensibility:* support for additional environment parameter measurement probes and control parameters is easily added. The measurement probes require the ability to activate them and support sampling and logging of the derived values. Adding support for new control parameters, e.g., video resolution scaling, implies extending the remote control protocol and requires to reconfigure the codec in real-time which is already present in the platform as explained earlier. Hence, the main task for the researcher to focus on is situated at the experiment design. The parameter space sampling scheme must be refined for the test scenario, such that the influence of the new parameter can be correlated to previous test results and the experiment time can be kept within certain boundaries. Analysis of the results is also to be reconsidered as the derived results take more parameters into account.

It should be clear that the implemented platform fulfills the requirements described in Section 3.1, providing the technology and flexibility to configure a wide range of real-time subjective assessment experiments for interactive applications, as well as the evaluation of algorithms that perform real-time adaptations to optimize the execution of interactive applications under constrained situations (beit under network or hardware constraints).

5 Platform Evaluation

5.1 Platform performance evaluation: methodology

For the evaluation of the performance of the platform implementation, the focus is mainly on the interactivity, i.e., the additional latency between the user input and the related graphical result caused by the platform must be acceptably low. Furthermore, reaching the targeted frame rate is an important requirement. For both evaluation criteria, instrumentation was provided into the software. For monitoring the achieved frame rate, straightforward measurement was provided by timing the method call delivering the decoded video to the display.

For the interactivity measurements, multiple interception points were introduced in order to obtain a breakdown of the delay components. The interception points are presented in Fig. 2, according to the action sequence needed to present a graphical response to user input. At these interception points, timestamps are taken, and all timestamps are logged at idle times, i.e., at the server side when an encoded video frame has been transmitted to the viewer, at the viewer side when a decoded frame has been delivered on screen and at the test application side when the X-server or OpenGL *draw* commands have been executed.

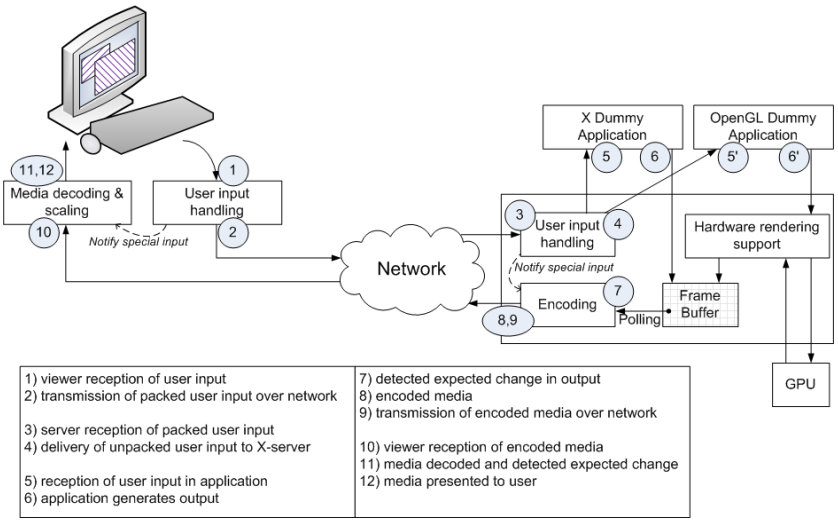


Fig. 2 Interception points where timestamps are taken, enabling a breakdown of the interactivity evaluation.

Table 2 Hardware configuration used for experiments

Server	Intel®Core^{TM}i7 CPU 920 @ 2.67 GHz, 6 GB RAM
Graphics Card	NVidia GeForce GTS 250, 128 CUDA cores
Server and viewer co-located on one physical machine	

To actually measure this delay, a dummy application was deployed on the platform, changing pixel colors on the display in response to a keystroke. This application was instrumented using the interception points discussed above. By correlating keystrokes and graphical updates, the incurred latency is evaluated. The application was implemented in OpenGL, to include the virtualization of the hardware rendering in the measurements. The keystrokes are generated at random time intervals using the *xdotool* library. With the obtained breakdown, it is possible to evaluate the overhead related to the end-to-end latency, caused by the thin client approach and by the virtualization of the hardware rendering of graphics.

For the experiments performed in this paper, the platform was installed on the machine of which the specifications are listed in Table 2. The viewer and server were deployed on the same machine. No network degradations were required for the experiments.

5.2 Platform performance evaluation: results

The implemented platform has been evaluated in terms of a breakdown of end-to-end latency incurred by the system, and the achieved frame rates. We also evaluated the feasibility of recording the graphical output of interactive sessions in real-time.

5.2.1 End-to-end latency

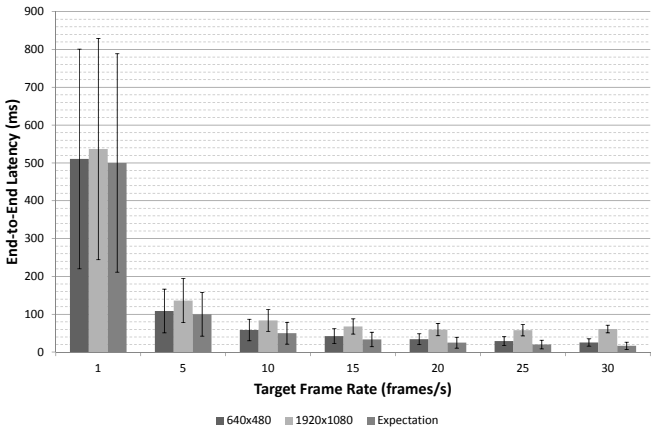


Fig. 3 End-to-end latency for varying base resolutions. The average latency slightly increases with higher resolutions mainly because of increased encoding complexity. The expectation presents the theoretical latencies involved with polling (i.e., a uniform distribution with mean value of half an inter-frame duration).

Figure 3 presents measurements of latency between user input and presentation of the graphical results on the screen, measured using the interactivity testing tool presented above. For this experiment, 5000 samples were taken for different target frame rate values and different screen resolutions. The figure shows that the average latency closely approaches the expected values of half an inter-frame duration (i.e., $\frac{1}{2} \times \frac{1}{TargetFrameRate}$), as the content is delivered at the given target frame rate. It is also visible that for the larger screen resolution, the latency increases due to the increased encoding time. Furthermore, we observe that the standard deviation is roughly independent of resolution and hence of encoding complexity. This standard deviation is inherently the highest for lower target frame rates, since in these cases the frame buffer is polled at a low frequency and thus the receipt of user input can happen in a relatively large time interval, and correlates to the theoretical standard deviation of a uniform distribution between zero and the inter-frame duration[1].

Table 3 and Table 4 present the breakdown of the reported end-to-end latencies, respectively for resolutions 640 × 480 and 1920 × 1080, with the timespan indications adhering to the interception points presented in Fig. 2. The timespan 6-7 represents the polling delay, of which the averages and standard deviations match the expected values from the uniform distribution as explained before. The major differences between the two resolutions are found in timespans 7-8 (media encoding time), 9-10 (network transmission of the encoded content) and 10-11 (media decoding). Timespan 7-8 indicates the average encoding time for one frame: at 640×480 resolution the mean encoding time is around 4.30 ms, while for 1920×1080

[1] For example, for a frame rate of 5 fps, the inter-frame rate is 200 ms. The average polling latency is therefore uniformly distributed between 0 ms and 200 ms, yielding an average value of 100 ms and a standard deviation of 57.7 ms.

Table 3 Average end-to-end latency breakdown measurement results from the interception points presented in Fig. 2, for resolution of 640 × 480, standard deviations reported between parentheses. All results are expressed in ms, contributions below 1 ms are omitted for clarity of the table.

Timespan	Frame rate				
	1	5	20	25	30
6-7 (Polling)	$501.88^{(290.58)}$	$99.76^{(57.90)}$	$24.69^{(14.32)}$	$19.88^{(11.60)}$	$16.43^{(9.46)}$
7-8 (Encoding)	$4.38^{(0.51)}$	$4.26^{(0.56)}$	$4.30^{(0.52)}$	$4.24^{(0.61)}$	$4.26^{(0.55)}$
9-10 (Network)	$1.19^{(31.60)}$	$1.56^{(16.45)}$	$1.84^{(8.92)}$	$1.75^{(7.80)}$	$1.60^{(6.79)}$
10-11 (Decoding)	$1.92^{(0.48)}$	$1.70^{(0.59)}$	$1.71^{(0.60)}$	$1.69^{(0.58)}$	$1.70^{(0.59)}$
1-12 (Total)	$510.69^{(290.55)}$	$108.74^{(57.88)}$	$34.04^{(14.39)}$	$29.065^{(11.63)}$	$25.44^{(9.61)}$

Table 4 Average end-to-end latency breakdown measurement results from the interception points presented in Fig. 2, for resolution of 1920 × 1080, standard deviations reported between parentheses. All results are expressed in ms, contributions below 1 ms are omitted for clarity of the table.

Timespan	Frame rate				
	1	5	20	25	30
6-7 (Polling)	$500.15^{(292.53)}$	$99.32^{(58.25)}$	$23.85^{(14.76)}$	$18.45^{(13.09)}$	$15.82^{(12.67)}$
7-8 (Encoding)	$18.66^{(1.70)}$	$18.37^{(1.59)}$	$18.39^{(1.64)}$	$18.40^{(1.70)}$	$18.50^{(1.74)}$
9-10 (Network)	$7.63^{(85.70)}$	$8.33^{(39.45)}$	$6.86^{(17.54)}$	$9.66^{(17.86)}$	$15.18^{(17.87)}$
10-11 (Decoding)	$7.12^{(2.06)}$	$7.04^{(2.10)}$	$7.25^{(2.13)}$	$8.21^{(2.72)}$	$8.55^{(2.90)}$
1-12 (Total)	$536.64^{(292.50)}$	$136.12^{(58.15)}$	$59.35^{(16.18)}$	$57.77^{(15.02)}$	$61.07^{(10.24)}$

resolution this encoding time increased to about 18.40 ms. A clear difference is also found in timespan 9-10, covering the latency caused by network transmission of the encoded media. This latency increases with the resolution (from about 1.70 ms to around 8 ms) as the encoding efficiency is considered of the same order, leading to increased data volume to be transmitted, hence the additional network latency. Due to the use of the Transmission Control Protocol (TCP) for network transmission, occasional retransmits cause the standard deviation to be significantly high. Timespan 10-11, dedicated to decoding, will increase with higher resolutions as well from 1.70 ms to 8 ms. Considering a frame rate configured high enough (e.g., 25 or 30 fps), we see that for Full HD resolution, the end-to-end latency keeps well under 80 ms which has been found as the boundary for unnoticeable end-to-end latency for dynamic interactive application scenarios such as games [6, 7]. However, we also note that the sequential execution of encoding the frame and transmission caused the inability to acquire 30 fps for Full HD resolution, since the latencies incurred by these two actions, 18.50 ms and 15.18 ms, are together already higher than 33 ms. Parallelization of media encoding and network transmission will enable Full HD resolution support over 50 fps.

5.2.2 Achieved frame rate

The frame rates that can be achieved are presented in Fig. 4. For these measurements, the application *glxheads* was executed fullscreen, drawing random triangles at a rate above 100 fps. The results show that the targeted video encoding frame rates are achieved except for high resolutions, such as Full HD (1920 × 1080), for which our test machine could not manage to deliver the requested frame rate in the configured high quality. The standard deviation depends on the complexity of

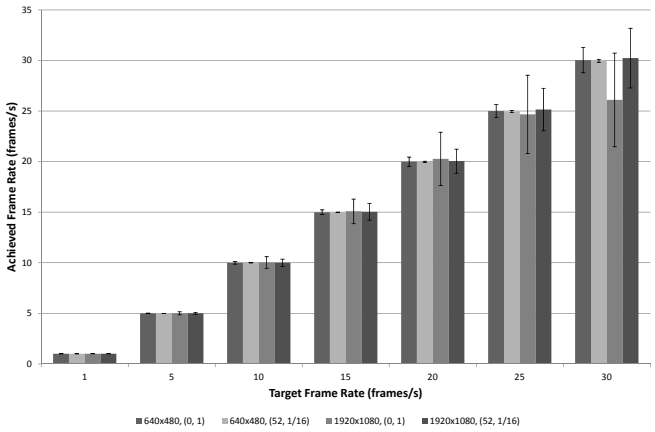


Fig. 4 Achieved frame rates for varying base resolutions with quantization parameter and resolution scaling configurations indicated between the brackets respectively. Setting (0, 1) correlates to QP 0, lossless compression, and no resolution scaling. Setting (52, 1/16) correlates to lossy compression, and a resolution scaling to 1/16 of the original resolution before encoding. The stability of the achieved frame rate is influenced by the complexity of the content.

the encoding operation. This can be noticed between the original resolutions, and between the different values for the (qp, resolution scale) pair. Low values of these parameters indicate high quality encoding, which is more complex and computing intensive. The value (0, 1) stands for no degradation from these parameters, hence perfect quality encoding is requested which is more complex than the (52, 1/16) setting for the parameters. More specifically, this value corresponds to a very lossy QP setting of 52, and a rescaling of the screen content to one sixteenth of the original before encoding. The figure also shows that the standard deviation increases with the complexity of the encoding. This is due to the implementation of the encoding loop, that rather aggressively compensates for variations in achieved frame rate. To ensure statistically relevant conclusions are drawn from the measurements, averages and standard deviation are computed over 600 samples.

5.2.3 Session recording

For recording purposes, the disk throughput plays an important role. If the test machine is equipped with a solid-state drive, the graphical updates can be written to that drive in raw format, i.e., 3 Bytes per pixel. For a resolution of 1920×1080 and at a rate of 30 frames per second, the needed throughput towards the solid-state drive is (1920×1080) pixels/frame $\times$ 3 B/pixel $\times$ 30 frames/s = 186.624 MB/s. We have measured an effective throughput of 248 MB/s to the ext4 file system using the Linux command line tool *dd*, showing that raw capture of the graphics stream in Full HD resolution at 30 frames/s is feasible. When recording the encoded streams (i.e., not the raw data) to disk, even higher resolutions and frame rates can be easily supported.

6 Use Case Evaluation

To show the applicability of the presented platform for subjective assessment of interactive media quality, the use case of interactive streaming of a video game has been explored. For this case, the sensitivity of the user to frame rate drops and visual quality degradation with respect to the interactivity experience is evaluated. Based on the results of this experiment, algorithms can be designed that mitigate bandwidth variations by configuring frame rate or visual quality while minimizing the impact on QoE. We have selected the race game *VDrift: Open Source Racing Simulator* [25], and focus our study on two parameters, i.e., frame rate and quantization parameter (QP).

For this use case, the experiment consists of two parts. First, the platform is used to quantify network load generated by the remote application execution as a function of the configurable parameters frame rate and QP. The purpose of this first part of the experiment is to constitute a benchmark that can be created automatically, and enables to estimate the effect of configuration decisions on the generated network traffic. Second, a subjective test designed to map the interactivity experience to different frame rates and QP settings is performed, by way of acquiring a Mean Opinion Score (MOS) from a sufficiently large set of users that rate their experience using questionnaires. For this experiment, the same setup is used as presented in Section 5.

The results of these experiments are presented in the following subsections. Finally, using these results, we propose an algorithm evaluation methodology that could be applied in a second subjective assessment round.

6.1 Automated benchmarking

It is important to execute a reproducible benchmark scenario in order to establish the sensitivity of the resulting bandwidth with respect to the configurable parameters. For this particular race game application, we selected a reproducible test scenario that causes similar graphical output for the different test settings.

Using a non-interactive application for the benchmarking relaxes the issues related to generating the user input for recreating the test scenario. We have created a benchmark with the non-interactive application *glxspheres*, one of the test applications that come with VirtualGL. This application exhibits similar graphical behavior as the selected racing game, i.e., the program ensures different content to be drawn on the screen at high frame rate. We have recorded the generated network traffic and computed the compression ratio expressed as relative bandwidth compared to the highest quality configuration. We have computed a mean difference of 0.2% relative bandwidth between the non-interactive application benchmark and the interactive application benchmark that requires a considerably larger effort to obtain. The results of the non-interactive application benchmarking are presented in Fig. 5. In this benchmark, where perfect similarity of the content nature is guaranteed over the varying configurations, the different relative bandwidth trends of frame rate and QP are prominent.

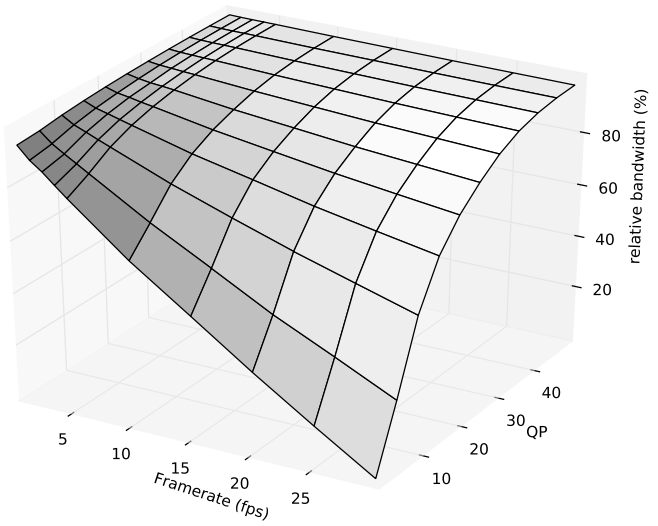


Fig. 5 Output of the platform: benchmark of effects of QP parameter and frame rate on bitrate for the glxspheres benchmarking application. (Resolution 1280 × 1024)

Table 5 Frequency table of test subjects' game and race game experiences (counted over 21 test subjects).

	Daily	Weekly	Monthly	Yearly	Never
Game Experience	4	6	6	5	0
Race Game Experience	0	2	7	11	1

6.2 Subjective assessment of playability of the race game

6.2.1 Subjective assessment setup

To ensure the used hardware is able to support the requested settings, the screen resolution is configured to 1024×768 pixels. The frame rates are configured between 30 and 5 fps, in steps of 5 fps. The set of QP values that is supported ranges from 0 (lossless encoding) to 45 (lossy encoding) in steps of 5. Twenty-one users from the authors' faculty are involved in all tests, with differing expertise in gaming as shown in Table 5, and all with a networking and software engineering background. We also performed a post-experiment screening of our test subjects using the methodology described in Annex V of the VQEG HDTV report [26] in order to detect outliers in our subjective data. This methodology is based on the linear Pearson correlation coefficient and rejects a subject's quality scores in case the correlation with the average of all the other subjects' quality ratings drops below the acceptability threshold. After this correlation test, nineteen of the subjects were retained as valid. The method for user input was restricted to keyboard only. The viewing distance was about 1 meter for all users. During all experiments, achieved frame rate and bitrate are logged to verify consistency.

Table 6 MOS scores and interpretation used in the subjective experiment.

Score	Interpretation
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

6.2.2 Subjective assessment design

When conducting subjective assessment experiments with respect to interactive applications, specific attention must be paid to providing time to the users to get acquainted with the controls of the application they need to interact with. In our case, the selected application has been deliberately chosen for its gentle learning curve. Since our study focuses on gaming experience, influence could be found in that the subjects get to know the controls, the track and the response model of the car better. We should avoid that the tested user evolves his gaming skills during the experiment (thereby consistently rating scenes later in the sequence different from earlier scenes). For our experiment, we have constituted the following scenario:

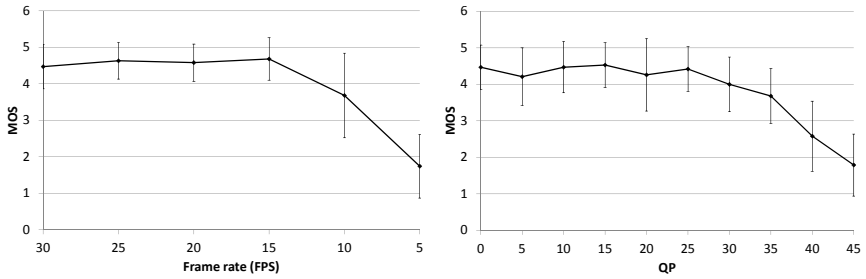
- Three minutes of free racing with non-degraded quality setting around the single track and car that are used throughout the experiment. During this acclimatization, we assisted the test subjects to ensure full understanding of the application. A printout of the needed controls was made available for reference throughout the test.
- Subsequently, two configurations were loaded, for which 40 seconds of racing is alternated with scoring the experienced playability along the scale presented in Table 6. The users entered their scores via the keyboard.
- After these two exercise scoring rounds, the actual assessment with the selected, randomized configurations was performed, using the same methodology of 40 seconds racing interleaved with scoring. For the scoring, one single question was asked: *"How well were you able to play the game, irrespective of visual quality or perceived artefacts?"*, allowing the subjects to focus on one specific aspect in the experiment.
- After completion of the experiment with the complete configuration set, we asked the subjects to give comments with the open question *"Could you comment shortly on which aspects you find most and/or least important concerning the playability of the racing game you just played?"*.

Contrary to the benchmarking method presented earlier, subjective assessment of the effect of tunable parameters, i.e., frame rate and QP for the current study, on the perceived interactivity of the application cannot be automated as such, since it actually requires test subjects to give their opinion about the settings. Therefore, in practice, it is unrealistic to explore all combinations of all possible parameter settings. Instead, we choose to vary one parameter while fixing all other to their highest possible value. Additional samples were selected to cover the configuration space. In total, we tested 33 configurations, that were presented in random order to the user. No network impairments were inflicted making the users effectively rate the raw effect of the configuration settings without experiencing side-effects such as network delays or occasional bandwidth variations.

Table 7 Frequency table of answers given to the open question *"Which aspects do you find most and/or least important concerning the playability of the racing game?"* (counted over 21 test subjects).

Answer	Frequency
Visual quality	5
Visibility of obstacles / corners	8
Reaction speed / Lag	10
Frame rate / Shockiness	12
Frame rate is more important than visual quality	11

6.2.3 Subjective assessment results



(a) Varying frame rate with QP fixed to 0. (b) Varying QP with frame rate fixed to 30 frames per second.

Fig. 6 MOS scores over 19 test subjects, for varying one configuration parameter with all other parameters fixed at their highest possible value.

Figure 6(a) presents the MOS scores acquired over the nineteen accepted subjects, for varying frame rate and a QP fixed at zero. From the figure, it is clear that for the race game under study, the users score the gaming experience of the game equal when the frame rate varies between 30 and 15 fps. When decreasing the frame rate below 15 fps, the user experience is hampered. These findings were verified using a paired t-test, applying a significance threshold $p = 0.05$. For Fig. 6(b), where QP is varied with frame rate fixed at 30 fps, a similar trend is noticed. For QP values between 0 and 25, no statistically significant difference in playability of the race game was found. A breakpoint is found around QP 25 as a significant difference is found in comparison with the measurements for QP 30. These findings were also verified using a paired t-test, with a significance threshold $p = 0.05$. Note that no data was gathered about the noticeability of the degradations, but that the user only scores the playability. Hence, the statistical equivalence of configurations only reports on the user acceptance of degradations, it does not exclude that the users see a difference between them.

From the comments given by the 21 test subjects to the open question *"Which aspects do you find most and/or least important concerning the playability of the racing game?"*, summarized in Table 7, we have distilled that over 50% has spontaneously reported that to them, frame rate is more important for the playing experience than still-frame quality. Also, about 50% reported frame rate and shockiness, and reaction speed to be of key importance. Only 5 of the users reported that

Table 8 Average MOS scores and standard deviations from the 19 accepted test subjects. The top left corner indicates the least lossy configuration, the bottom right corner correlates to the configuration with most information loss. The gray area indicates statistically equivalence, as computed using the t-test ($p = 0.05$).

	Frame rate					
QP	30	25	20	15	10	5
0	4.47 (0.61)	4.63 (0.50)	4.58 (0.51)	4.68 (0.58)	3.68 (1.16)	1.74 (0.87)
5	4.21 (0.79)	4.42 (0.69)				
10	4.47 (0.70)	4.37 (0.83)	4.50 (0.62)			
15	4.53 (0.61)	4.53 (0.61)		4.39 (0.92)	3.28 (1.07)	
20	4.26 (0.99)		4.53 (0.77)			2.17 (1.04)
25	4.42 (0.61)		4.47 (0.70)		3.37 (0.90)	
30	4.00 (0.75)			3.89 (0.94)	3.47 (0.93)	
35	3.68 (0.75)	3.53 (0.70)		3.68 (0.82)		
40	2.58 (0.96)				2.21 (0.92)	
45	1.79 (0.85)		1.95 (0.97)		1.47 (0.61)	1.11 (0.32)

visual quality has a big influence, while 8 persons commented specifically that the visibility of corners and obstacles is a lower bound for visual quality. From these comments, we derive that, for the race game, frame rate has a higher impact on the QoE than visual quality. The fact that for games, users tend to find frame rate to have a larger impact on QoE than visual quality is in accordance to the experimental results presented in [1]. This finding is validated through the MOS scores given, as we found in the t-tests for Fig. 6(a) and Fig. 6(b) that the statistical significance from the breakpoint off was much higher in the frame rate dimension than the visual quality dimension. Also from the MOS scores, it is visible that in the frame rate dimension, the descent is steeper than for the visual quality. The average MOS scores of the accepted test subjects are listed in Table 8, with the grey area indicating configurations with statistically insignificant score differences as computed with the t-test with threshold $p = 0.05$.

We emphasize that these results are currently only validated for the race game, and are expected to differ for other applications or other contexts. For example, the current task for the user in the test was to drive laps around a track, without specifically focusing on speed. If the task would have been to beat a lap or speed record, the context would change making readability of the speedometer and lap times more important, possibly increasing the influence of visual quality on QoE. Also, other types of games, e.g., shooter games, might exhibit more stringent requirements on frame rates and visual quality, while other types of interactive applications, such as image editing or text typesetting might be less sensitive to these factors for QoE.

6.3 Subjective algorithm evaluation

From the results obtained from the benchmarking and the subjective test, algorithms can be designed that mitigate bandwidth variations by controlling the frame rate and the visual quality while optimizing interactive QoE. Such *Graceful degradation* algorithm would measure the used bitrate and find the needed compression ratio with the highest expected quality score. To evaluate algorithms, a second round of subjective tests with the platform should be organized, selecting

a number of target bitrates and gathering QoE scores using the designed algorithm and dummy algorithms for comparison. Examples of algorithms are *Closest matching compression* that is purely bitrate based and solely takes the compression efficiency into account to decide about parameter settings, or *Maximize frame rate* that aims at keeping the frame rate as high as possible at all times, sacrificing the visual quality if needed.

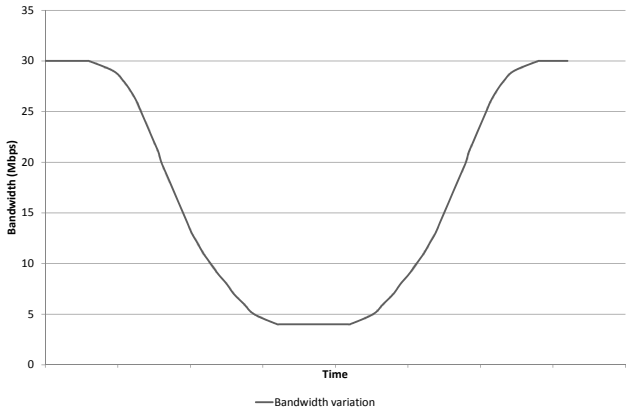
We obtained a model of the MOS and compression ratio as a function of the parameters frame rate and QP, by fitting the MOS scores and the results from benchmarking presented earlier. Both models exhibit a coefficient of determination R^2 of 0.95, which confirms an adequately accurate fit. Using these models, we can estimate the outcome of the selected algorithms and find suited target bandwidth configurations to design a meaningful second subjective test iteration to verify the performance of the algorithms. We propose to vary the bandwidth during the test, e.g., along a variation scheme as presented in Fig. 7(a), and to query the user for their opinion on the QoE for the enabled algorithm. Figure 7(b) presents the MOS scores that are expected to be obtained by the algorithms proposed above, and are derived using the fitted models as follows:

1. Given the available bandwidth, the needed compression ratio is computed.
2. From the compression ratio model, a reverse lookup using this needed compression ratio yields a set of possible configurations of QP and frame rate.
3. From this set, one configurations is chosen according to the algorithm. The *Closest matching compression* algorithm selects the discrete combination of parameters that has the lowest positive Euclidean distance to the desired compression ratio. The *Maximize frame rate* algorithm chooses the configuration with highest possible frame rate. The *Graceful degradation* algorithm selects the configuration with the highest MOS (i.e., computed from the MOS model).
4. Finally, the MOS scores for these configurations are computed using the MOS model.

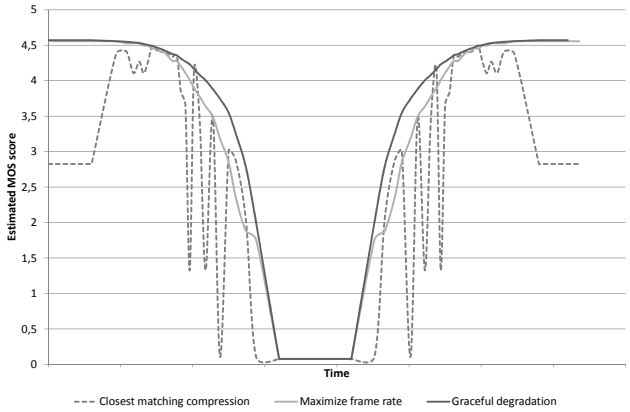
In Fig. 7, it can be seen that the *Closest matching compression* algorithm is expected to underperform due to its purely naive nature. Furthermore, we predict that with the *Graceful degradation* algorithm, the best interactive QoE can be delivered, or stated otherwise, the algorithm can deliver better QoE for given bandwidths. For instance, if a MOS of 3.5 is considered as a lower threshold, the *Maximize frame rate* algorithm reaches this limit at 11 Mbps, while the *Graceful degradation* algorithm reaches it at 8 Mbps.

7 Conclusions

This paper presents a novel platform that allows structured subjective user testing of a broad range of interactive applications. Current subjective testing methods are mainly based on prerecorded video streams, which limit the applicability to interactive applications since interactivity cannot be captured in advance without loss of assessment potential afterwards. Therefore, on-the-fly adaptation of parameters is supported in the platform, such that test users interact with the applications in real-time, to evaluate the effect of the varying parameter settings on the user experience. In addition to the frequent focus on network conditions in literature, the influence of other parameters such as frame rate and visual quality of media



(a) Bandwidth variation over time, to apply in a second subjective test iteration.



(b) Expected MOS scores obtained by different algorithms in response to bandwidth variation.

Fig. 7 Estimated outcome of a second subjective test iteration. Estimations are based on models fitted over the MOS scores and compression ratios obtained in the first test iteration and automated benchmarking phases of the evaluation.

codec settings, as well as integral algorithms that control various parameters in real-time can be assessed in a structured, fully controllable and reproducible manner. Examples of use cases that can be subjectively assessed using the platform are given in the paper, with associated deployment configurations to perform them.

The platform is based on thin client computing, which allows network modeling due to the separation of user input and graphical presentation from application

logic. The architecture is shown to fulfill the requirements listed: (i) ability to model the network, (ii) real-time controllable media encoder parameters, (iii) a modular approach, (iv) live interaction with the applications and (v) reproducibility of the experiments.

Evaluation of the prototype implementation indicates the feasibility and applicability of the platform for subjective assessment of interactive applications. The results show that with adequate hardware, the target frame rates can be achieved with expected deviations that depend on the encoding complexity of the media. The end-to-end latency between user input and presentation of the resulting application output to the user falls close to the expectations. Furthermore, initial results of a subjective assessment of an interactive multimedia-oriented application are presented, that show different trends of the impact of visual quality and frame rate on QoE for a race game i.e., a prevalence of frame rate over visual quality is found.

Directions for future work include application of the platform for the design of new subjective testing methods and objective metrics that take interactivity into account. Furthermore, subjective experiments will be performed that show the dependency of user perceived quality on characteristics of the application and the performed task. Also, various newly designed algorithms, e.g., mitigating network delay through graphics caching, will be evaluated using the platform.

References

1. Chang, Y.C., Tseng, P.H., Chen, K.T., Lei, C.L.: Understanding the performance of thin-client gaming. In: Communications Quality and Reliability (CQR), 2011 IEEE International Workshop Technical Committee on, pp. 1–6 (2011). DOI 10.1109/CQR.2011.5996092
2. Chen, K.T., Huang, P., Lei, C.L.: How sensitive are online gamers to network quality? Commun. ACM **49**(11), 34–38 (2006). DOI 10.1145/1167838.1167859
3. Chen, K.T., Tu, C.C., Xiao, W.C.: OneClick: A framework for measuring network quality of experience. In: INFOCOM 2009, IEEE, pp. 702 –710 (2009). DOI 10.1109/INFCOM.2009.5061978
4. Chen, K.T., Wu, C.C., Chang, Y.C., Lei, C.L.: A crowdsourceable QoE evaluation framework for multimedia content. In: ACM Multimedia, pp. 491 – 500 (2009)
5. Claypool, M.: Motion and scene complexity for streaming video games. In: Proceedings of the 4th International Conference on Foundations of Digital Games, FDG '09, pp. 34–41. ACM, New York, NY, USA (2009). DOI 10.1145/1536513.1536529
6. Claypool, M., Claypool, K.: Latency and player actions in online games. Commun. ACM **49**(11), 40–45 (2006). DOI 10.1145/1167838.1167860
7. Dick, M., Wellnitz, O., Wolf, L.: Analysis of factors affecting players' performance and perception in multiplayer games. In: Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games, NetGames '05, pp. 1–7. ACM, New York, NY, USA (2005). DOI 10.1145/1103599.1103624
8. Engel, K., Sommer, O., Ertl, T.: A framework for interactive hardware accelerated remote 3d-visualization. In: Proc. TCVG Symp. on Vis. (VisSym), pp. 167–177 (2000)
9. International Telecommunication Union: ITU-T Recommendation P.920. Interactive test methods for audiovisual communications (2000). URL http://www.itu.int/
10. International Telecommunication Union: ITU-T Recommendation P.910. Subjective video quality assessment methods for multimedia applications (2008). URL http://www.itu.int/
11. Jarschel, M., Schlosser, D., Scheuring, S., Hossfeld, T.: Gaming in the clouds: QoE and the users' perspective. Mathematical and Computer Modelling (2012). DOI 10.1016/j.mcm.2011.12.014
12. Jovic, M., Hauswirth, M.: Measuring the performance of interactive applications with listener latency profiling. In: Proceedings of the 6th international symposium on Principles

and practice of programming in Java, PPPJ '08, pp. 137–146. ACM, New York, NY, USA (2008). DOI 10.1145/1411732.1411751
13. JSON: JavaScript Object Notation (JSON) (2012). URL http://www.json.org/
14. Jumisko-Pyykkö, S., Utriainen, T.: A hybrid method for quality evaluation in the context of use for mobile (3d) television. Multimedia Tools and Applications **55**(2), 185–225 (2011). DOI 10.1007/s11042-010-0573-4
15. Kohler, E., Morris, R., Chen, B., Jannotti, J., Kaashoek, M.F.: The click modular router. ACM Trans. Comput. Syst. **18**, 263–297 (2000). DOI 10.1145/354871.354874
16. Kuipers, F., Kooij, R., De Vleeschauwer, D., Brunnstrm, K.: Techniques for measuring quality of experience. In: Wired/Wireless Internet Communications, *Lecture Notes in Computer Science (LNCS)*, vol. 6074, pp. 216–227. Springer Berlin / Heidelberg (2010)
17. Lai, A.M., Nieh, J.: On the performance of wide-area thin-client computing. ACM Trans. Comput. Syst. **24**(2), 175–209 (2006). DOI 10.1145/1132026.1132029
18. Moorthy, A., Bovik, A.: Visual quality assessment algorithms: what does the future hold? Multimedia Tools and Applications **51**(2), 675–696 (2011). DOI 10.1007/s11042-010-0640-x
19. Nieh, J., Yang, S.J., Novik, N.: Measuring Thin-Client Performance using Slow-Motion Benchmarking. ACM Trans. Comput. Syst. **21**(1), 87–115 (2003). DOI 10.1145/592637.592640
20. Quax, P., Monsieurs, P., Lamotte, W., De Vleeschauwer, D., Degrande, N.: Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game. In: Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games, NetGames '04, pp. 152–156. ACM, New York, NY, USA (2004). DOI 10.1145/1016540.1016557
21. Ries, M., Svoboda, P., Rupp, M.: Empirical study of subjective quality for massive multiplayer games. In: Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on, pp. 181–184 (2008). DOI 10.1109/IWSSIP.2008.4604397
22. Santiago, P., Ignasi, I., Elisa, M., Antonio, M.J.: TRUE: an online testing platform for multimedia evaluation. In: In Proceedings of the 2nd Int. Workshop on EMOTION: Corpora for Research on Emotion and Affect at the 6th Conference on Language Resources & Evaluation (LREC) (2008)
23. Stegmaier, S., Magallón, M., Ertl, T.: A generic solution for hardware-accelerated remote visualization. In: Proc. TCVG Symp. on Vis. (VisSym), pp. 87–96 (2002)
24. Tolia, N., Andersen, D., Satyanarayanan, M.: Quantifying interactive user experience on thin clients. Computer **39**(3), 46 – 52 (2006). DOI 10.1109/MC.2006.101
25. VDrift: VDrift Open Source Racing Simulator, version 2011-10-22. http://www.vdrift.net
26. Video Quality Experts Group: Report on the validation of video quality models for high definition video content. Tech. rep. (2010). URL http://www.its.bldrdoc.gov/vqeg/projects/hdtv/. http://www.vqeg.org/
27. VirtualGL: VirtualGL (2012). URL http://www.virtualgl.org
28. Wattimena, A.F., Kooij, R.E., van Vugt, J.M., Ahmed, O.K.: Predicting the perceived quality of a first person shooter: the Quake IV G-model. In: Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games, NetGames '06. ACM, New York, NY, USA (2006). DOI 10.1145/1230040.1230052
29. Yang, C., Niu, Y., Xia, Y., Cheng, X.: Performance analysis of interactive desktop applications in virtual machine environment. The Chinese Journal of Electronics **17**(2), 242 – 246 (2008)
30. Yoo, S.H., Yoon, W.C.: Modeling users' task performance on the mobile device: PC convergence system. INTERACTING WITH COMPUTERS **18**(5), 1084–1100 (2006). DOI 10.1016/j.intcom.2006.01.003
31. Zeldovich, N., Chandra, R.: Interactive performance measurement with VNCplay. In: Proceedings of the annual conference on USENIX Annual Technical Conference, ATEC '05, pp. 54–64. USENIX Association, Berkeley, CA, USA (2005). Http://suif.stanford.edu/vncplay/