An adaptive control system to deliver Interactive Virtual Environment content to handheld devices

An Adaptive Control System to Deliver Interactive Virtual Environment Content to Handheld Devices

Gianluca Paravati · Andrea Sanna ·
Fabrizio Lamberti · Luigi Ciminiera

Abstract Wireless communication advances have enabled emerging video streaming applications to mobile handheld devices. For example, it is possible to display and interact with complex 3D virtual environments on mobile devices that don't have enough computational and storage capabilities (e.g. smart phones, PDAs) through remote rendering techniques, where a server renders 3D data and streams the corresponding image flow to the client. However, due to fluctuations in bandwidth characteristics and limited mobile device CPU capabilities, it is extremely challenging to design effective systems for streaming interactive multimedia over wireless networks. This paper presents a novel approach based on a controller that can automatically adjust streaming parameters basing on feedback measures from the client device. Experimental results prove the effectiveness of the proposed solution in coping with bandwidth changes, thus providing high Quality of Service (QoS) in remote visualizations.

Keywords video streaming to mobile devices · closed-loop controller · QoS · remote visualization

G. Paravati (✉) · A. Sanna · F. Lamberti · L. Ciminiera
Dipartimento di Automatica e Informatica,
Politecnico di Torino, C.so Duca degli Abruzzi 24,
10129, Torino, Italy
e-mail: gianluca.paravati@polito.it

A. Sanna
e-mail: andrea.sanna@polito.it

F. Lamberti
e-mail: fabrizio.lamberti@polito.it

L. Ciminiera
e-mail: luigi.ciminiera@polito.it

1 Introduction

The combination of advances in wireless communication together with the rapid evolution and growing popularity of mobile handheld devices has led to useful applications within the field of 3D graphics and virtual reality. For instance, virtual guiding malls, multiplayer games and collaborative virtual environments are emerging applications now available on thin devices, i.e. with low processing and storage capabilities. The increased capability of wireless connectivity allows the display of graphically attractive environments on devices with limited hardware capabilities through remote rendering techniques where a server renders 3D data and streams the corresponding image flow to the mobile client [1, 2]; the user on the client side can interact with the 3D scene by sending navigation commands to the server.

This work presents the design of a novel feedback-based controller in the context of image streaming systems with interactivity constraints; the controller is designed to automatically adapt streaming parameters to both bandwidth fluctuations and device characteristics such as maximum supported resolution and processing capabilities. The proposed technique works at the application level and it can be applied to any kind of client device; however, in this work, we specifically focus on common handheld devices, such as PDAs and smart phones because of their limited computational capabilities. Preliminary results indicate the effectiveness of the proposed approach, although it is only a starting point for future work investigating control-based remote rendering.

The main features of the proposed system include: an increase in visualization quality through smooth changes in streaming parameter values, the adaptation of streaming parameters to bandwidth fluctuations and adjustments to end device capabilities. The proposed control technique is general, and as such, it can be used in any image compression-based streaming scenario. The presented solution is also compared with a technique based on optimization methodology [6].

The remainder of the paper is organized as follows. Section 2 reviews previous works related to the management of streaming parameters and control theory approaches. Section 3 presents the details of the proposed control framework. Experimental results are discussed in Section 4.

2 Related work

Real-time and interactive streaming to handheld devices over variable bandwidth channels must address different Quality of Service (QoS) issues, ranging from network features (such as bandwidth fluctuations and channel latency) to mobile device capabilities (such as receiver decoding performance). The concept of QoS addressed in this paper refers to the issues pertaining the achievement of interactive frame rates and low latencies for interactive remote visualization applications. These issues are investigated in depth in [3]; to deploy effective remote visualization applications, a few QoS requirements must be met. If these requirements cannot be met, the user will be probably unable to get the expected results. The QoS requirements that are most directly related to remote visualization are low delay, high throughput/bandwidth and low latency. Motion-JPEG (M-JPEG) has proven to be an effective means for obtaining very low latencies and low processor overhead at the expense of increased bandwidth [4, 5]. In fact, M-JPEG parameters like resolution, frame rate and image quality (all of which determine the bandwidth occupation of the streaming flow) can be combined and tuned to suit network characteristics. The relationship among these parameters can be modelled as in [6]:

$$f = \frac{B}{w \cdot h \cdot C_d \cdot \frac{1}{C_r}} \tag{1}$$

where f is the achievable frame rate, w and h denote image resolution (R), B is the currently available bandwidth, C_d is the color depth in terms of bits per pixel, and C_r is the compression ratio of an image with respect to the same uncompressed picture (i.e., C_r is strictly related to the image quality Q). A graph providing a formal representation of remote visualization parameters can be defined; in this graph, each node represents a possible pair of values R and Q (a configuration of parameters) that can be used to code the flow of still images to be streamed from the server to the client device. Two or more configurations are close when at least one of their parameters varies little. The edges of the graph represent possible transitions between two close configurations; they are dynamically re-built according to a metric that periodically takes into account bandwidth occupation, user requirements (for instance, the desired frame rate) and device throughput. The graph is then "travelled" to reach the node (i.e., the configuration) that provides the best bandwidth occupation and matches user requirements.

Control theory approaches to performance management for computer systems such as Internet web servers, databases and storage systems, have been successfully applied in the past. In [7], a study on the admission control for an Internet web server is presented in which a linear-parameter-varying (LPV) approximation for the modeling of the dynamic relationship from the request rejection ratio to the response time for the admitted requests is used. The main characteristic of an LPV controller is the capacity to control non-linear dynamics of state variables using different working conditions resulting from external agents influencing the system. Since in our case, the dynamic relationship between image resolution and needed bandwidth is non-linear, this approach initially seemed to be suitable to control the modification of the above parameters. However, in a M-JPEG streaming scenario the image quality dynamics introduce an unpredictable impact on the size of the compressed image to be delivered. Indeed, unfortunately, when we change the quality setting, the resulting size of the compressed image cannot be known a priori. This reflects directly on the bandwidth occupation, and therefore, this is the main drawback that prevented us from using LPV approximation in our streaming scenario.

Since working conditions in interactive streaming to mobile handheld devices can differ depending on the available channel bandwidth, attention has been focused on adaptive control techniques [8, 9] to develop a controller able to modify its control parameters according to state variations. Adaptive control techniques estimate the behavior of the system through linear regression algorithms, where functionality is based on step-by-step refinement in order to asymptotically reach a set of parameters able to represent the unknown initial system. In the context of video streaming, these methods are not suitable because of the large fluctuations in the parameters; an estimator based on these techniques cannot identify an asymptotically stable system.

After establishing that the application of complicated control techniques does not favor the solution of the problem, a sort of PID-based (Proportional Integral Derivative) controller was considered; its robustness and reliability are particularly useful to control a system characterized by unpredictable fluctuations and to correctly exploit the feedback channel of the system. To achieve performance specifications, gain-scheduling techniques were used. Gain scheduling [10] is based on the idea of using different (a-priori performed) calibrations in different circumstances, thereby realizing a parameter calibration system able to adapt to the state of the system. This approach (see Section 4) provided smoother transitions than the optimization methodol-ogy proposed in [6], thus providing users with a better visualization experience.

3 The control algorithm

The design of the proposed controller aims to take advantage of automatic control techniques to concurrently tune all parameters involved in a remote rendering scenario (i.e., resolution, image quality and frame rate) without any a priori knowledge about the precise effects caused by altering these parameters. The main requirements identified in the design phase of the controller are: sensitivity to feedback measures, robustness to non-linearity, independence to the quality of the underlying network and optimal usage of available resources.

The control system is sketched out in Fig. 1. The controller is fully implemented on the server side, meaning that the client has only to compute its own device throughput and to periodically feed this information to the streaming server. It is worth remarking that in this paper, the term robustness is related to the degree to which the controller can function correctly in the presence of uncertainties affecting our model of the system.

The control system mainly refers to the theoretical bandwidth occupation $TBO(t)$, which depends on the current encoding parameters and can be computed by reversing Eq. 1. The measured output of the system on the client side is the device throughput $DT(t)$. The architecture of the controller system is characterized by a single input and two separate outputs. The input is a bandwidth error $\Delta B(t) = DT(t) - TBO(t)$. The outputs are resolution (R) and quality (Q) factors; they are used by the rendering server to create, compress and stream through the network an image representation of the 3D scene to the mobile device. The bandwidth error $\Delta B(t)$ aims to continuously consider the spread between the visualization flow generated by the rendering server and the amount of the flow that the mobile client is able to process. The device throughput depends both on network conditions and device capabilities; whenever the device throughput is lower than the theoretical bandwidth occupation, this indicates that the mobile device is not able to receive, decode and display all the visualization flow sent by the server. As a consequence, the controller will try to reduce the encoding parameters to minimize the wasting of bandwidth.

The controller addresses bandwidth control by trying to balance the system around a target frame rate, which is either preconfigured or provided by the user. Indeed, the value of the target frame rate directly influences $TBO(t)$; a higher value for the target frame rate requires a higher theoretical bandwidth occupation. The concept of a target frame rate has been introduced to

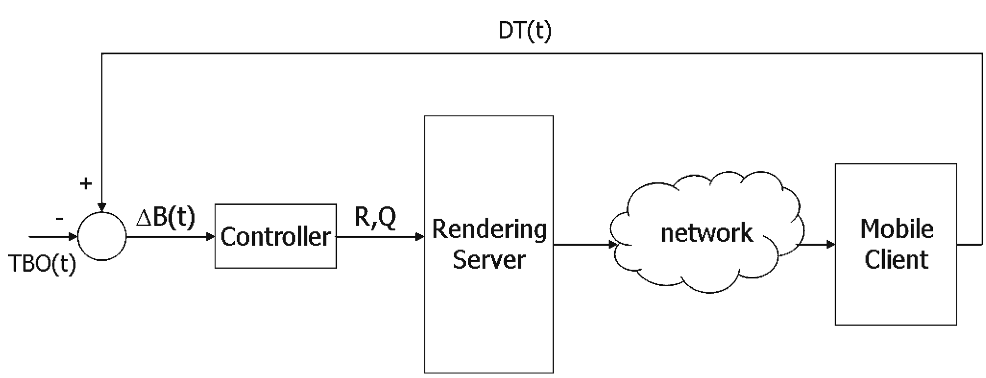


Fig. 1 The logical control system

allow for the external control of the parameters. Depending on the task to be performed, users can be more interested both in motion smoothness and interactivity or, more generally, in a higher level of visual details. In the first case, the system should respond by shifting the trade-off toward higher values of frame rate, whereas in the latter case, the system should favor higher values of resolution and quality independent of frame rate. In the proposed method, the user can influence the controller's behavior by acting on the target frame rate. It is worth remarking that regular users are not always supposed to understand what constitutes a reasonable value for a target frame rate; however, it is possible to easily combine some preconfigured values in rules describing the user preferences (e.g., the use of a slider can hide technical details to users, still allowing them to express their preference in terms of smoothness or level of details). In the proposed framework, the actual frame rate of the client (as computed using Eq. 1) i s continuously compared to the target frame rate; the controller tries to minimize the error or difference between these two values. The system variables (i.e., quality and resolution) are controlled so as to undergo changes that are proportional to the error measured as input. That is the system changes the parameters of the encoded flow of images in such a way to reach the desired frame rate in order to exploit the minimum between the current available bandwidth, which depends on network status, and the current device's processing capabilities. Indeed, the feedback measure of the controller is always narrowed down by the smallest of these two values.

The reciprocal interaction between the controlled variables (i.e., R and Q) causes unpredictable nonlinear fluctuations in the required bandwidth to be used for in the transmission of the data stream to the client, thus resulting in a hard design phase. This issue can be tackled by isolating the relationship that binds each variable to the image size and thus establishing the degree of variation needed to increase or decrease the image size (and the bandwidth occupation) to a determined quantity.

The basic idea is to subdivide the available bandwidth in different portions and assign them to each system variable; in this way, we apply a sort of superposition principle to compute the variation of each system variable in consecutive steps. The mathematical relations that govern the growth or the reduction of a system variable, such as the bandwidth changes, allow us to individually set each variable by fixing the other parameters. A mathematical relationship that links the bandwidth quantity to be filled (or released) and a resolution increment (or decrement) can be defined as:

$$\Delta R = \frac{\Delta B}{B_R^{\mathrm{up}} - B_R}; \qquad (2)$$

where ΔR is the amount of the increment or decrement in resolution, ΔB is the bandwidth error measured by the controller, B_R and ${B_R}^{\mathrm{up}}$ are the bandwidth to be used with the current resolution index and the bandwidth to be used with the immediately upper resolution index, respectively. The proportionality between resolution and bandwidth is clearly visible from Eq. 2; however, Eq. 1 makes explicit the relationship between frame rate and bandwidth, showing how a resolution change can be reflected in a frame rate change through a proportionality coefficient k_R:

$$\Delta R = k_R \cdot \Delta f. \qquad (3)$$

The controller also addresses another system variable represented by the compression quality. The identification process of the control relationship for this variable is more complicated mainly because it is difficult to evaluate, even approximately, a mathematical relationship between image quality and compression factor, which changes reflect to the compressed image size through an inverse proportionality relationship. Thereby, we decided to exploit the controller feedback measure, varying the quality level at each loop and then correcting the excessive bandwidth repercussions if necessary. The idea is to subsequently control a variable that cannot be controlled a priori because of its unpredictability. Since a precise bandwidth variation implies a precise frame rate variation, a direct relationship between image quality and frame rate is established. In fact, these variables are not linearly dependent because the incidence of the frame rate is determined by the compression factor C_r, which has a non-deterministic dependence on image quality. Such a relationship can be modeled as:

$$\Delta Q = k_Q \cdot \Delta f. \qquad (4)$$

The instability both of the available bandwidth and of the device throughput makes unstable the input reference error ΔB of the controller; thus, it vacillates rather than asymptotically approaching to a stable equilibrium value. Each instant can be characterized by different working conditions. To manage this instability, an adaptive control approach can be used. It allows us to find some specific coefficients for the control equations; these coefficients are relative to external system variables and the current state of the system. The k_R coefficient can be expressed as:

$$k_R = \frac{k_p}{k_B} = \frac{k_p}{B_R^{\text{up}} - B_R}; \tag{5}$$

where k_p is a proportional constant of a PID-based system, and k_B is an adaptive parameter that depends on the current resolution. In this way, k_R is derived using a value that is also used to compute the bandwidth occupied by a change in resolution, which is represented by k_B, and a value used by the system to tune the bandwidth quantity to be assigned to the resolution, which is represented by k_p. As a result, the input bandwidth is subdivided into a portion that is assigned to the resolution variable. As the coefficient k_p increases, a greater portion of bandwidth is reserved to allow for a change in resolution, thereby enhancing the sensitivity of the system to bandwidth fluctuations and its reaction to reference tracking. The unpredictable nature of bandwidth fluctuations, particularly in wireless connections, renders the integral-derivative component of a PID-based controller useless at present. Indeed, the accumulation of past errors (i.e., the integral component) and the prediction of future errors (i.e., the derivative component) are not currently taken into account by this technique.

The subdivision of the bandwidth across the system variables is an important aspect for the controller. Indeed, the two variables of resolution and image quality cannot be precisely tuned at the same time because of their reciprocal interactions; nevertheless, it is possible to act individually on each of them through the relationships between these variables and the reference parameter, i.e., the target frame rate. Thus, the control phase is split into two stages. In both stages, it is possible to act on a single variable by proceeding in cascade, with a different bandwidth as an input at each stage and fixing all other parameters to modify only the variable of interest. At each stage, the input bandwidth is used to modify the variable of interest. In this way, the system can be characterized as a SISO model. In the first stage, the bandwidth error ΔB is received as input and used to regulate resolution; the new resolution value causes a change in the theoretical bandwidth occupation and the achievable frame rate computed based on the current parameters. This resolution is thus used as input of a second stage to re-compute the bandwidth error. The second stage receives the new bandwidth error as input, which is used to regulate quality according to Eq. 4. T h e coefficient values have been found by an empirical trial and error procedure during a calibration phase, which leads to the estimations $k_p = 0.6$ and $k_Q = 0.4$ as the most suitable values. The meta-algorithm of the model is reported as follows to clarify the description of the controller.

Meta-algorithm

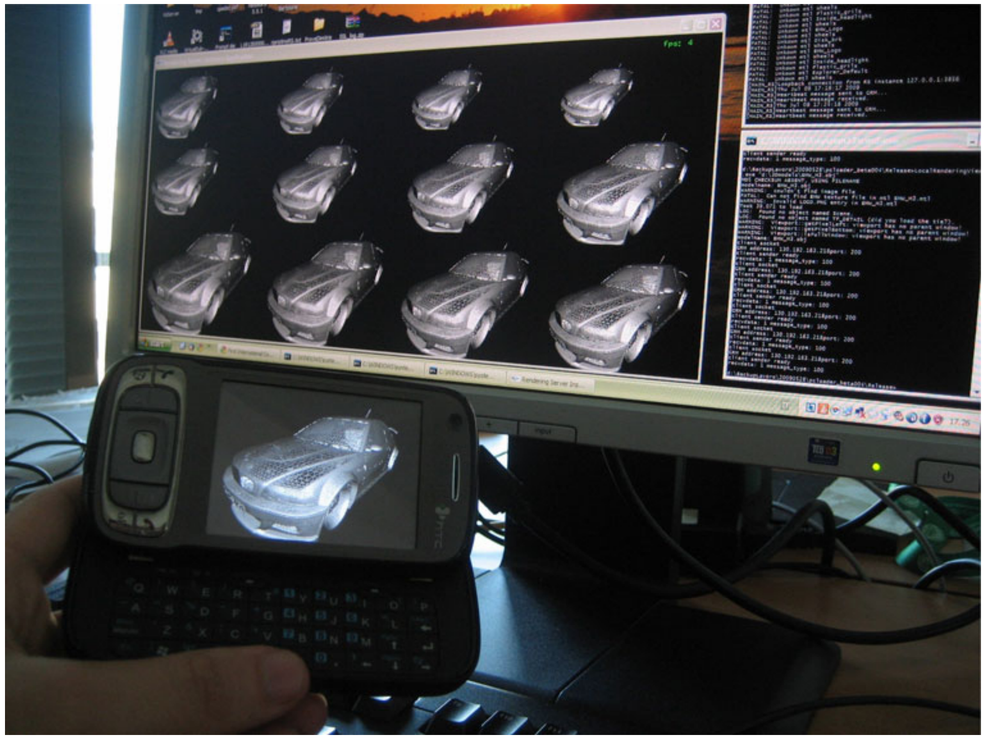
1: Compute the compression ratio of the last image
2: Evaluate the current frame rate of the device using Eq. 1
3: Compute the difference between the current frame rate and the target frame rate 4: Evaluate $B_R{}^{\text{up}}$ by reversing Eq. 1 using the current encoding parameters
5: Compute k_R using Eq. 5 with $k_p = 0.6$
6: Update the resolution using Eq. 3
7: Compute the new theoretical bandwidth occupation by reversing Eq. 1 and using the updated encoding parameters
8: Evaluate the new bandwidth error as the difference between the measured throughput and the new theoretical bandwidth occupation
8: Compute Δf using Eq. 1
9: Compute the quality increment using Eq. 4 with $k_Q = 0.4$
10: End

4 Tests and results

The proposed controller has been implemented and tested in a remote rendering scenario. The rendering server runs on a Dual-Core AMD Opteron CPU 2.60 GHz workstation equipped with 3.50 GB of RAM and with an NVIDIA Quadro FX 3500 graphics card; it was developed using C++ language and is based on the OpenSG library. The client program runs on a HTC TyTN II smart phone, which is connected to the rendering server through an 802.11g wireless access point. The client program was developed using J2ME (Java Micro Edition), as it provides a flexible environment for applications running on mobile and other embedded devices. Figure 2 shows a remote rendering session; the server (shown in the background) is in charge of rendering the 3D scene and streaming a M-JPEG flow to the "thin" client (shown in the foreground).

Figure 3 shows the behavior of the controller under varying system conditions; the monitored parameters

Fig. 2 A remote 3D rendering session on a TyTN II smart phone



are (from top to bottom): resolution, image quality, actual frame rate reached by the mobile device and measured throughput (i.e., the feedback feature). Figure 3a shows the ability of the controller to react to changes in network conditions. The system is configured to maintain a target frame rate of $FR_{\text{target}} = 15$ fps throughout the entire test. Indeed, the client application allows the user to express his/her preference in terms of frame rate; a higher frame rate enhances motion smoothness and impacts on perceived interactivity. During the first part of the simulation (i.e., the period between $t = 0$ and $t = t_1$) the system works in stationary conditions to exploit the maximum throughput of the device ($DT \approx$ 60 KB/s). A trade-off between resolution and quality was reached in order to maintain the target frame rate. At time $t = t_1$, a maximum bandwidth limitation was imposed on the server side ($BW_{\text{limit}} = 40$ KB/s) to simulate a network bottleneck; after the frame rate on the client side rapidly dropped down (from $FR \approx$ 15 fps to $FR \approx 9$ fps), the controller reacted by gradually reducing the parameter values. While changing parameters, the controller continuously compared the target frame rate with the effective frame rate. After $t = t_2$, the controller leaded the system to a different steady state, again reaching the target frame rate and thus mitigating the effects of the bandwidth bottleneck previously introduced. If the target frame rate was higher, parameters would be continually reduced until the target frame rate was reached again. Figure 3b d e - picts the capacity of the controller to follow a specified target frame rate. Initially, the system maintained a target frame rate equal to 15 fps during the period from $t = 0$ and $t = t_3$ and the system was steady around high values of resolution ($R = 288 \times 216$ pixels) and quality ($Q \approx 80$) parameters. At time $t = t_3$ the target frame rate was set to 25 fps through the client GUI. During the period between $t = t_3$ and $t = t_4$ the controller re-acted by gradually reducing resolution (R $= 176 \times 132$ pixels) and quality ($Q \approx 28$). After the transition, the system reached a different steady state with the new frame rate value; despite the growth in the number of frames per second received, the device throughput is lower than before because images are more com-pressed, thus indicating that performance is limited by the decoding capabilities of the mobile device.

In this work, a set of experimental tests of user experience were carried out to evaluate the relationship of the designed controller with QoS parameters. A group of 19 subjects was asked to carry out two sets of tests on the designed system and later answer a questionnaire aimed at collecting feedback on system performance. Each user was individually trained to know how to use features of the remote visualization application running on an HTC TyTN II mobile device. The trainer performed two sessions (i.e., with and without the proposed controller), illustrating to the selected

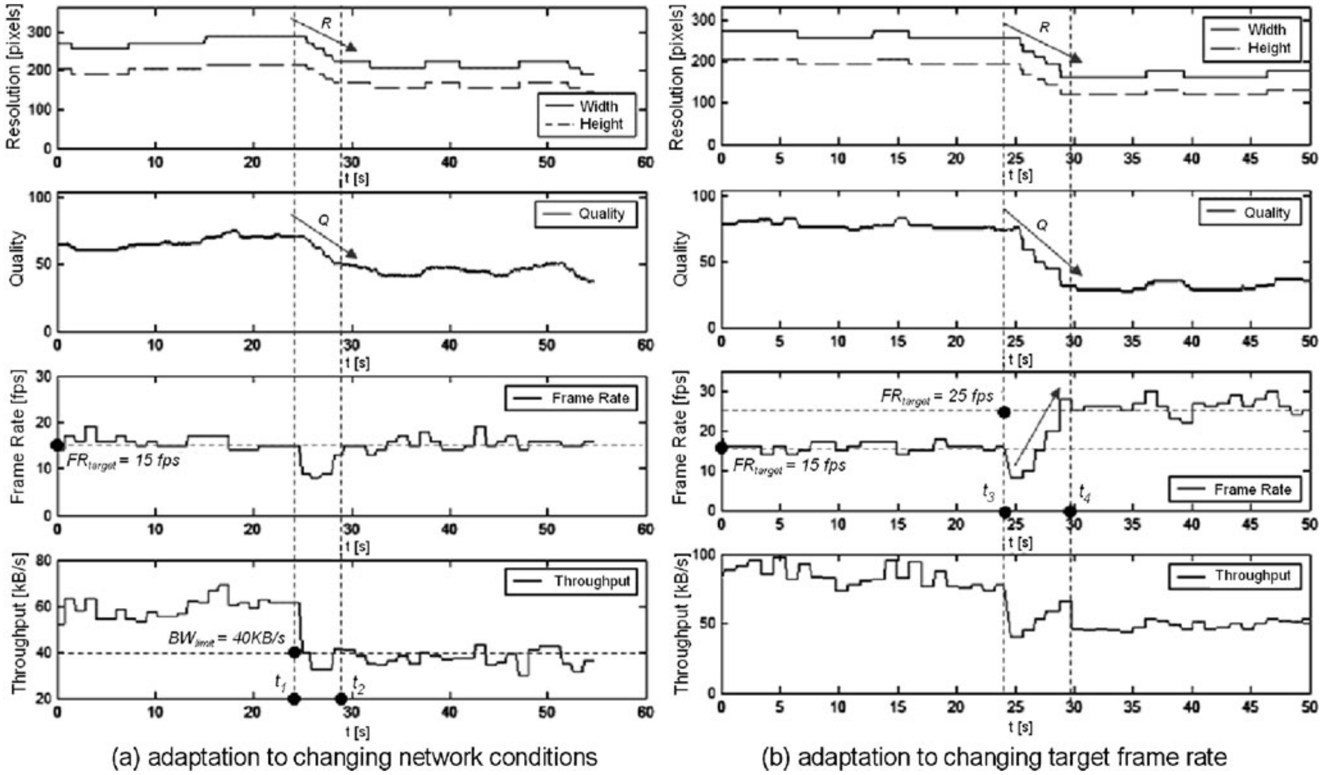


Fig. 3 Performance on a TyTN II smart phone: (**a**) adaptation to changing network conditions and (**b**) adaptation to changing target frame rate

subjects the aspect and behavior of the user interface; in particular, the possibility of changing the target frame rate was emphasized. After the training phase, each user was allowed to use the device alone. Each user was asked to perform the two tests by navigating the 3D scene; during the first test the controller was disabled, while during the second test it was enabled. No time limitation was set, and all the users completed the tests within five minutes).

During the first test, the image stream characteristics did not change; in particular, encoding parameters were configured to match the maximum device resolution with the best image quality. This choice led to a poor frame rate. During the second test, the controller was enabled and users were allowed to change the target frame rate. In the questionnaires, users were then asked to assign marks from 1 to 10, with 1 being the lowest evaluation and 10 the highest evaluation, with respect to the perceived performance of the visualization tool both with and without the controller. Half of the testers started using the system without the controller and then switched to the controlled system; the other half of testers started with the controller to counterbalance the experiments.

A statistical analysis of the experimental data was carried out to compare the performance of the remote visualization with (C_{ON}) and without (C_{OFF}) the controller. Since variances were unknown and small samples were taken, the t-statistic was used; a paired t-test was performed by testing the hypothesis that the mean of differences between each pair of scores μ_t is $\mu_{C_{\mathrm{ON}}} - \mu_{C_{\mathrm{OFF}}} = 0$. A level of significance $\alpha = 0.05$ was used for computing the reference t-value for comparison based on the distribution table. According to this statistical analysis, the t-statistic is greater than the reference t-value, and the null hypothesis can be rejected. Since $\mu_t > 0$, the controlled system results outperform the uncontrolled one. This constitutes effective proof of the relationship between the user experience and QoS parameters.

4.1 Control vs. optimization

The proposed methodology was also compared with the technique presented in [6]; this method is based on a dynamic graph in which nodes represent possible configurations of parameters, i.e., R and Q, and arcs represent the transitions between two close

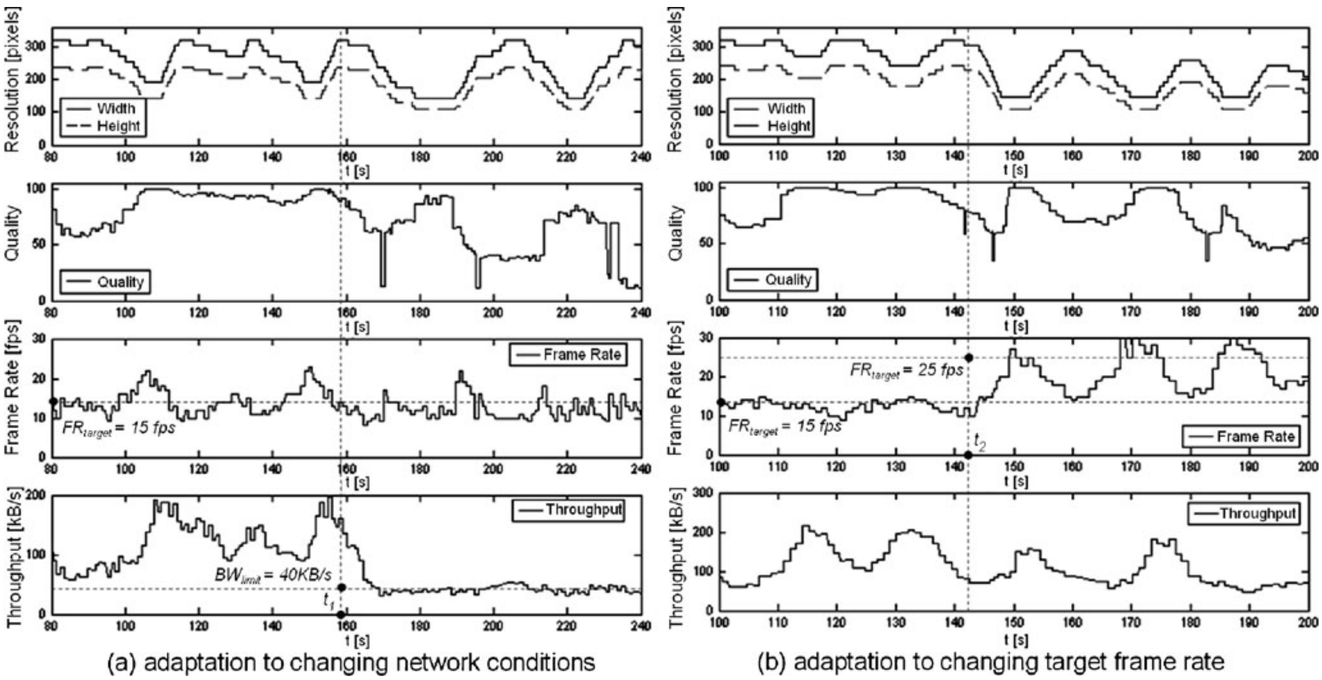


Fig. 4 Performance of the optimization technique [6] on a TyTN II smart phone: (**a**) adaptation to changing network conditions and (**b**) adaptation to changing target frame rate

configurations, i.e., two nodes whose parameters vary little.

Figure 4 shows the behavior of the optimization technique presented in [6]. These results were gathered under the same conditions and the same parameters as the tests performed for the proposed controller (see Fig. 3). The optimization technique allows us to arrange resolution, quality and frame rate in the order of subjective importance to the user. Also, to make a fair comparison between the two methods, the most important remained the frame rate, whereas resolution and quality were given the same weight. During the overall simulation in Fig. 4a, the target frame rate was set to 15 fps. At time t_1, the network bandwidth was artificially constrained to 40 KB/s at the server side using a network limiter application, thus simulating a network bottleneck. Based on the graphs in Fig. 4a, it is possible to see that the monitored parameters are continuously subject to fluctuations. In this specific case, the resolution decreases when the quality increases and viceversa; the parameters continue to oscillate and this behavior leads to instability issues in the system. From a qualitative point of view, visualization is negatively influenced by these consecutive changes. However, this situation can be avoided by properly configuring a controller, such as the proposed controller; indeed, given the current available bandwidth, an optimization-based algorithm can automatically configure a set of parameters, thereby enabling the best exploitation of the available bandwidth. However, in this case, the modification of the first parameter has a direct influence on the second parameter. If the resolution decreases, the same available bandwidth can be exploited with a higher value of quality and vice versa, potentially triggering drift problems as outlined in Fig. 4. However, a closed-loop controller can provide finer control and quick reaction times even to slight changes, thereby minimizing the drift problem. This same behavior can be observed for the second test depicted in Fig. 4b. Initially, the frame rate was set to 15 fps; at time t_2, the target frame rate changed to 25 fps. Although the mean value of resolution and quality decreased to provide a higher value for the frame rate, they continuously oscillated and they did not reach a stable value, thus resulting in annoying effects, e.g. repeated scaling, on the screen.

5 Conclusions

Due to bandwidth-demanding characteristics and limited mobile device capabilities, it is extremely challenging to design effective systems for streaming interactive multimedia over wireless networks for handheld devices. This paper presents a controller able to automatically adjust M-JPEG streaming parameters; the solution was implemented and tested in a remote rendering scenario in which a smart phone interactively displayed complex 3D virtual environments despite its limited computational capabilities.

Although the proposed solution is tailored for mobile handheld devices, it is general and can be applied to any interactive streaming scenario; moreover, this control technique can be applied to video compression schemes such as MPEG. The controller is implemented on the server side; it exploits a feedback measure from clients and thus, it is able to continuously adapt to changing network conditions, different device throughput capabilities and different interaction requirements. The control mechanism mainly focuses on reaching a target frame rate then modifies other parameters such as resolution and image quality. However, the same mechanism can be used but it needs to be rearranged to follow a different parameter, such as resolution or quality.

Future studies should aim to investigate different control methodologies that could further enhance the system robustness. For instance, the use of robust control methods can be investigated to address the uncertainties of the model developed for this work.

Acknowledgement The authors wish to thank Riccardo Friaglia for his invaluable help in the implementation of the framework.

References

1. Pazzi RWN, Boukerche A, Huang T (2008) Implementation, measurement, and analysis of an image-based virtual environment streaming protocol for wireless mobile devices. IEEE Trans Instrum Meas 57(9):1894–1907
2. Quax P, Geuns B, Jehaes T, Lamotte W, Vansichem G (2006) On the applicability of remote rendering of networked virtual environments on mobile devices. In: International conference on systems and networks communications, pp 16–16
3. Stegmaier S, Diepstraten J, Weiler M, Ertl T (2003) Widening the remote visualization bottleneck. In: Proc. third int'l symp. image and signal processing and analysis, pp 174–179
4. Endoh K, Yoshida K, Yakoh T (2008) Low delay live video streaming system for interactive use. In: IEEE International conference on industrial informatics, pp 1481–1486
5. Nishantha D, Hayashida Y, Hayashi T (2004) Application level rate adaptive motion-JPEG transmission for medical collaboration systems. In: The 24th international conference on distributed computing systems workshops, pp 64–69
6. Paravati G, Sanna A, Lamberti F, Ciminiera L (2008) A novel approach to support quality of experience in remote visualization on mobile devices. In: Eurographics 2008 short paper proceedings, pp 223–226
7. Qin W, Wang Q (2007) An LPV approximation for admission control of an internet web server: identification and control. Control Eng Pract 15:1457–1467
8. Lu Y, Abdelzaher T, Lu C, Tao G (2002) An adaptive control framework for QoS guarantees and its application to differentiated caching services. In: The 10th IEEE international workshop on quality of service, pp 23–32
9. Wu K, Liljia DJ, Bai H (2005) The applicability of adaptive control theory to QoS design: limitations and solutions. In: The 19th IEEE international parallel and distributed processing symposium
10. Slotine JJ, Li W (1991) Applied nonlinear control. Prentice Hall, Englewood Cliffs, New Jersey 07632