A Novel Approach to Support Quality of Experience in Remote Visualization on Mobile Devices

G. Paravati, A. Sanna, F. Lamberti, L. Ciminiera

Politecnico di Torino, Dipartimento di Automatica e Informatica, Italy

Abstract

This paper proposes a novel approach to support Quality of Experience (QoE) in remote visualization on mobile devices. Image resolution, frame rate, compression ratio, color depth, and device throughput are simultaneously considered in order to provide users valuable visualization experiences. User requirements are used by a QoE manager to estimate values for the above parameters, thus providing an optimal usage of the network bandwidth. Experimental results show how the proposed methodology can efficiently support remote visualization on mobile devices such as Personal Digital Assistants and smartphones allowing the user to obtain satisfactory visualization sessions.

Categories and Subject Descriptors (according to ACM CCS): I.3.2 [Computer Graphics]: Distributed/network graphics

1. Introduction

Although performance of handheld, and more in general, mobile devices have tremendously increased in the last years, several applications can only run on desktop machines. For instance, online applications for the mobile workers, providing them with advanced, personalised, content and context aware tools for search and retrieval of rich multimedia information can involve the visualization of highly complex 3D models that is out of the capabilities of mobile platforms.

The remote visualization paradigm is often used to overcome this problem. The application is running on a remote server machine and the (mobile) client device receives only a data representation as a flow of still pictures or as a video stream. Video streams allow to obtain better compression ratios but need a higher computational power to decode incoming information (not always available on mobile devices) and introduce a higher latency due non causal compression schemes. A flow of JPEG images is sent from the server to a client device in this work. Two basic approaches are known to implement remote visualization: (a) the entire desktop of the remote machine is sent to the client device (this solution does not well support small screen client devices) [RSFWH98] (b) ad-hoc solutions are designed and implemented to allow the client device to directly control the interface of the remote application and receive as result a certain area of the remote desktop selected as working area [LS07, SDWE03]. Both methodologies involve the transmission of a set of pixels from the server to the client; a wireless transmission channel is usually involved in remote visualization sessions with mobile devices.

The ultimate measure of a network and the services it offers is how subscribers perceive the performance. Quality of Experience (QoE) [EMP03, Ner03, NOK04] is the term used to describe this perception and how well subscribers think the services are usable. This paper tackles the issue of QoE within a remote visualization framework for mobile devices. There are several factors that interact to determine QoE, including cost, reliability, availability, usability, utility, and fidelity. If the QoE is high, then the user is satisfied, on the other hand, low QoE indicates that the user does not have a good experience of the service. QoE and QoS (Quality of Service) are two parameters used to measure the quality of a service. QoS is the ability of the network to provide a service with an assured service level, while QoE is how a user perceives the usability of a service and how he/she is satisfied. Quality of Service is intrinsically a technical concept. It is measured, expressed and understood in terms of networks and network elements, which usually has little meaning to a user. QoS is a subset of the overall QoE scope. Although a better network QoS in many cases will result in better QoE,

© The Eurographics Association 2008.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

fulfilling all traffic QoS parameters will not guarantee a satisfied user.

Delivering high QoE depends on gaining an understanding of the factors contributing to the user's perception of the target services, and applying that knowledge to define the operating requirements. This top-down approach reduces development costs and the risks of user rejection and complaint by ensuring that the device or system will meet user requirements. Although QoE is very subjective in nature, it is very important that a strategy is devised to measure it as realistically as possible.

This paper proposes a work in progress about a QoE framework design and implementation to support remote visualization of 3D objects on mobile devices. The QoE manager is part of a more complex architecture (project VICTORY `http://www.victory-eu.org/`) aimed to support users in search, retrieval, and visualization of 3D models over a Peer-to-Peer network. The main contribution of this work is a novel metric for estimation and management of parameters affecting QoE in remote visualization. At this stage of implementation, a remote application is able to: load a 3D model (most common 3D object format files are supported), accept a connection from a (mobile) client device, deliver to the client the frame buffer content compressed as a JPEG image, and receive back from the client a set of commands to navigate the scene (i.e. the user can roto-translate the displayed object). The QoE manager is able to concurrently consider a set of parameters: image resolution, frame rate, compression ratio, color depth, and device throughput. The measure of all these parameters would be too expensive, thus only a few of them are actually measured, while the best part of them are estimated by means of a metric. Moreover, the QoE manager receives from the user a sort of *goal function*: the user can specify the minimum frame rate to be guaranteed. The QoE manager will be able to dynamically choose an optimal set of parameters to provide the user a satisfactory visualization experience, that is it will be able to maximize the usage of the network bandwidth.

The paper is organized as follows: Section 2 describes the basic idea behind the QoE manager design. Section 3 presents examples and results and analyzes the accuracy of the proposed methodology. Finally, conclusions and future work are provided in Section 4.

2. Basic Idea

In order to assure good user experiences, data sent to the client have to be adapted to adhere to limitations of the client's terminal and network characteristics. Such adaptation could be, for instance, scaling the image resolution in order to fit on the terminal's screen. The resource negotiation will allow the determination and assessment of user QoE from individual quality of services specified or measured within the connection network. The key performance indicators for the remote visualization are basically:

- download bitrate (bandwidth);
- frame rate;
- resolution;
- compression ratio (affecting the image quality);
- color depth;
- command latency time for 3D content manipulation;
- processing speed;
- power consumption;
- security issues.

The last four issue are out of the scope of the QoE manager presented in this work. For instance, processing speed and power consumption are strictly related to each other and they depend on hardware characteristics. Command latency depends on the communication channel, while security issues are not directly related to the quality of a visualization session.

On the other hand, it is possible to correlate the other parameters by the following equation:

$$f(l,m) = \frac{B_{rx}^{client}}{w(l) \cdot h(l) \cdot Cd \cdot \frac{1}{Cr(l,m)}} \quad (1)$$

where $\mathbf{l}$ is the image resolution index, i.e. $\mathbf{l}$ identifies a set of rows in Table 1 having the same resolution ($\mathbf{w}(l)$, $\mathbf{h}(l)$), $\mathbf{m}$ is the quality of the image index (i.e. $\mathbf{m}$ is an index that denotes the JPEG quality and identifies a set of rows in Table 1 having the same quality), $\mathbf{f}$ is the received frame rate, $\mathbf{B_{rx}^{client}}$ the client bandwidth in reception (this term depends both on the communication channel and on the delay of the client in decoding, resize and display a compressed frame), $\mathbf{Cd}$ is the color depth in terms of Bits Per Pixel (BPP), and $\mathbf{Cr}$ is the compression ratio of an image with respect to the same uncompressed picture. For instance, images at a resolution of 640x480 pixels, 24 BPP, and a compression ratio of 20:1, need a bandwidth equal to 11Mbps if transmitted to 30 frames per second.

If the transmission bandwidth exceeds the average bandwidth of the client's connection, the user could receive a non smooth image sequence, thus it has been selected a target data rate based on the average download bandwidth of the client. Mobile user will be able to maximize perceived QoE by reaching a trade-off between image quality, image resolution, and frame rate, thus maximizing the bandwidth usage.

The QoE manager uses a table that correlates all these parameters to choose the best trade-off. An example is shown in Table 1. The first two columns list image resolution $\mathbf{w} \times \mathbf{h}$ (resolutions typically supported by PDAs and smartphones are considered, but the table could be extended in order to consider any resolution), the size $\mathbf{S}$ of the raw image (at a color depth of 24 BPP) is presented in the third column, the JPEG images quality $\mathbf{Q}$ is reported in the fourth column, compression ratios $\mathbf{Cr} = \frac{raw\_image\_size}{compressed\_image\_size}$ are listed in the fifth column, and the percentage growths of compression ratios $\boldsymbol{\Delta}_{\boldsymbol{Cr}}^{\boldsymbol{\%Q}}(l,m) = 100 \cdot \left(\frac{Cr(l,m)}{Cr(l,m-1)} - 1\right)$ with respect to the image, at the same resolution and compressed at a quality Q immediately higher, are in the last column.

© The Eurographics Association 2008.

Table 1: *Each row of the table represents a possible choice of parameters to be selected for a visualization session. Parameters w and h are expressed in pixels; Size S is expressed in bytes.*

w	h	S	Q	Cr	$\Delta_{Cr}^{\%Q}$
120	100	36000	100	9.66	-
120	100	36000	85	20.45	111.70
120	100	36000	70	24.42	19.41
120	100	36000	55	27.31	11.83
240	180	129600	100	17.11	-
240	180	129600	85	38.31	123.90
240	180	129600	70	47.52	24.04
240	180	129600	55	53.96	13.55
320	240	230400	100	19.24	-
320	240	230400	85	44.70	132.33
320	240	230400	70	56.21	25.75
320	240	230400	55	64.34	14.46
640	480	921600	100	32.33	-
640	480	921600	85	71.17	120.14
640	480	921600	70	87.75	23.30
640	480	921600	55	98.59	12.35

Information in Table 1 allows the QoE manager to "tune" parameters for a visualization session according to equation 1. A periodic network feedback mechanism supplies to the rendering server information concerning the average download bandwidth of the client. An estimation method able to compute all records of the table has been developed. Once all records are filled with measurements and estimations the QoE manager is able to determine the achievable frame rate for each entry of the table (i.e. each combination of QoE parameters). The user can negotiate the parameter priority; for instance, if a user wants to advantage perceived image quality by keeping above a certain threshold the frame rate, all the entries of the table that bring to reach a frame rate lower than desired can be discarted. Furthermore, an automatic mechanism has been implemented: the QoE manager receives from the user a sort of "goal function" (minimum frame rate), then it tries to automatically adjust the parameters, based on the device throughput (B_{rx}^{client}) and information about the current "status" of the rendered scene:

- Static view (the user does not browse the scene): the highest quality image to the detriment of frame rate is computed.
- Dynamic view (the scene is changing): the QoE manager tries to guarantee the frame rate choosen by the user switching between the other parameters while maximizing the bandwidth usage.

The problem to be tackled to support this strategy is the population of the table. The client bandwidth in reception B_{rx}^{client} can be periodically computed/updated and parameters of a visualization session have to be chosen in order to saturate it (i.e. the best QoE can be obtained maximizing the bandwidth usage); on the other hand, computing all records of the table could be an expensive task (at the remote server side) involving a bottleneck. A metric is proposed to overcome this problem. Records of the table are grouped in sets: sets at constant resolution ($w \times h$) and sets at constant quality (Q). Two indices l and m are used to index resolution and quality sets, respectively. For instance, $l = 0$ denotes the first four rows of Table 1, $l = 1$ denotes rows 5, 6, 7, and 8, and so on. On the other hand, $m = 0$ denotes rows 1, 5, 9, and 13, that is the rows where the quality index $Q = 100$. At this point, only a few records of the table are really measured; in particular, a set at constant resolution $l = kl$ and a set to constant quality $m = km$ are computed. Let us consider Table 1; only seven rows are really computed (in this case each set is composed by four records), while the other nine records are estimated. This metric can be applied to any set size: a greater number of resolutions as well as quality indices Q could be considered. The pseudo-code of the algorithm implementing the proposed metric is the following:

```
for(l=0;l<L;l++)
  for (m=0;m<M; m++)
  {
    if(l == kl) Cr already measured
    else if (m == km) Cr already measured
    else if (m > km) Cr as in equation (2)
    else (m < km) Cr as in equation (3)
  }
```

where **L** is the number of sets at constant resolution, **M** the number of sets at constant quality, and equations 2 and 3 are computed as:

$$Cr^{est.}(l,m) = Cr^{meas.}(l,km) \cdot \prod_{i=1}^{m-km} \left(1 + \frac{\Delta_{Cr}^{\%Q}(km+i)}{100}\right) \quad (2)$$

$$Cr^{est.}(l,m) = Cr^{meas.}(l,km) \cdot \prod_{i=1}^{km-m} \left(\frac{\Delta_{Cr}^{\%Q}(km-i+1)}{100} + 1\right)^{-1} \quad (3)$$

where $\boldsymbol{\Delta_{Cr}^{\%Q}}(m) = 100 \cdot \left(\frac{Cr(kl,m)}{Cr(kl,m-1)} - 1\right)$. The precision of the proposed metric (i.e. the difference between estimated and measured compression ratios) is discussed in Section 3.

3. Experimental Results and Remarks

The proposed approach was implemented and simulated on a variety of mobile devices ranging from smartphones to PDAs and Tablet PCs. The remote visualization scenario consisted of a workstation, with a dual-core AMD Opteron processor operating at a frequency equal to 2.60 GHz and with a NVIDIA Quadro FX 3500 graphic card, that acts as the rendering server and thin devices such as Nokia N93, Dell Axim X50V and Samsung Q1 Ultra Mobile PC acting as mobile clients. An example of visualization session is depicted in Figure 1. The server application, developed in C++ language using OpenGL libraries, accepts incoming connections and implements the remote visualization approach. Client applications are written in Java language

© The Eurographics Association 2008.

Figure 1: *An example of the remote visualization session running on a Nokia N93 smartphone.*

(J2ME for smarthphones and PDAs) and allows the user to display 3D models, send navigation commands to the rendering server, and control QoE parameters that affects performances. Client devices are connected to the rendering server via a WLAN 802.11b access point. The rendering server pre-computes different images at different resolutions in order to best fit client characteristics and needs. The user can change resolution and quality of the image interactively, thus changing visualization frame rate as a matter of fact. The client application measures and displays the real frame rate. In this way, it is possible to validate the correspondence between estimated parameters and the real ones. For example, the real frame rate chosen for the test depicted in Figure 1 was 13 fps, while the client bandwidth in reception in that moment was 76864 byte/s. Resolution was set to 320×240 pixels, JPEG quality was set to $Q = 85$ and the measured compression ratio was $Cr^{meas} = 37.96$. According to eq. 1, with this data it is possible to obtain a theoretical frame rate equal to $f = 12.66$ fps, thus confirming the measure. Changing JPEG quality to 70, the estimated compression ratio computed to fill Table 1 with the proposed method is $Cr^{est.} = 48.32$ and the estimated frame rate computed applying eq. 1 is equal to $f = 16.12$ fps with respect to the real measured value of 16 fps.

In order to check the precision of the proposed algorithm and compute the error committed in predicting compression ratios, simulations have been performed measuring the compression ratios for each individual set of parameters (i.e. for each line of the Table 1) for each frame generated by the rendering server. It has been calculated the mean value of the error between estimated and measured compression ratios $\overline{\varepsilon_{Cr}} = \sum \left| Cr^{est}(l,m) - Cr^{meas}(l,m) \right| / n$. On the average, $\overline{\varepsilon_{Cr}} = 0.88$ for each individual frame of the image sequence. Maximum and minimum errors for each combination of QoE parameters was $\varepsilon_{Cr}^{min} = 0.12$ and $\varepsilon_{Cr}^{max} = 2.16$.

4. Conclusions and Future Work

This paper presents a work in progress concerning a QoE manager to efficiently support remote visualization on mobile devices. A set of parameters affecting a visualization session are identified. These parameters allow to fill a table that enables the QoE manager of choosing the best trade-off in order to maximize the usage of the network bandwidth. The measure of all parameters would be a computationally intensive task, thus a few parameters are measured and the others are estimated by means of a proposed metric.

Future work will be aimed to tune and extend the proposed methodology in order to support collaborative sessions where several users (possibly using different types of devices and exhibiting different QoE requirements) can share the visualization of 3D scenes.

Acknowledgment

This paper is part of a work developed within the European Research Project VICTORY, "Audio-VIsual ConTent search and retrieval in a distributed P2P repositORY". VICTORY is funded under the Information Society Technologies (IST) thematic program of the European Commission's Sixth Framework Program.

References

[EMP03] Assuring qoe on next generation networks. *Empirix white paper* (2003). http://www.empirix.com/Empirix/Corporate/resources/resources+white+papers.html. 1

[LS07] LAMBERTI F., SANNA A.: A streaming-based solution for remote visualization of 3d graphics on mobile devices. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (Mar. 2007), 247–260. 1

[Ner03] NERGER P.: Managing quality of experience for mobile data services. 1

[NOK04] Quality of experience (qoe) of mobile services: Can it be measured and improved? *Nokia white paper* (2004). http://www.nokia.com/NOKIA_COM_1/About_Nokia/Press/White_Papers/pdf_files/whitepaper_qoe_net.pdf.

[RSFWH98] RICHARDSON T., STAFFORD-FRASER Q., WOOD K., HOPPER A.: Virtual network computing. *IEEE Internet Computing 2*, 1 (Jan. 1998), 33–38. 1

[SDWE03] STEGMAIER S., DIEPSTRATEN J., WEILER M., ERTL T.: Widening the remote visualization bottleneck. In *Proc. of the 3rd Int'l Symp. on Image and Signal Proc. and Analysis* (2003), pp. 174–179. 1

© The Eurographics Association 2008.