Avatar Mobility and Network Condition-aware 3D Game over Wireless Networks

Donghyeok Ho
Division of IT Convergence Engineering
Pohang University of Science and Technology
Pohang, Republic of Korea
positiveho@postech.ac.kr

Hyungnam Kim and Hwangjun Song
Department of Computer Science and Engineering
Pohang University of Science and Technology
Pohang, Republic of Korea
{hyungnam, hwangjun}@postech.ac.kr

***Abstract*—In this work, we present an avatar mobility and network condition-aware 3D game over wireless networks. The proposed system is designed to pursue an effective tradeoff between 3D rendering quality and compressed image quality. By considering user-preferred networking cost (i.e., data transmission charges), 3D rendering level and image compression parameters are determined dynamically in order to enhance the quality of service for interactive 3D game. The system is implemented using VirtualGL and an online open-source 3D game, PlaneShift, and examined using a mobile device connected to LTE and WiFi networks.**

***Index Terms*—3D game, avatar mobility, view distance adaptation, networking cost, rate-distortion model**

I. Introduction

With the growth and success of mobile applications, there is a fast growing interest in advanced multimedia applications at mobile devices that require considerable computational power and GPU capabilities, e.g., three-dimensional (3D) rendering, augmented reality, and speech recognition, etc. Especially, 3D rendering is not only computationally expensive; it can also impose severe challenges on the limited battery capacity of an always-on mobile device. Cloud computing technology has been a promising solution for less powerful mobile devices. In 2013, International Data Corporation (IDC) expects the cloud service market to grow at a Compound Annual Growth Rate (CAGR) of 23% from 2013 to 2018 [1]. Moreover, Juniper research estimates that revenue from mobile enterprise cloud-based applications and services will rise from nearly $2.6 billion in 2011 to $39 billion in 2016 [2]. In cloud gaming [3], a cloud server renders 3D content and compresses the rendered scenes as a video sequence. The resulting encoded video is then streamed to the mobile device through wireless networks. So far, a considerable amount of research efforts have been devoted to providing high-quality 3D rendering services based on the cloud platform. In [4, 5], an adaptive rendering algorithm is proposed, wherein multiple rendering factors such as view distance and texture detail are dynamically adjusted to satisfy constraints on communication and computation. The offline steps include developing a model in advance, considering the rendering richness and video quality. This model is then used online in real-time to determine the optimal rendering factors for a certain network bandwidth.

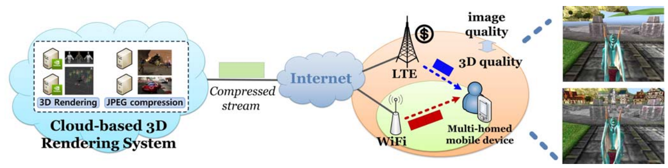


Fig. 1. Mobile cloud-based 3D game over wireless networks.

However, the simplified offline quality model has difficulties in supporting the 3D game of optimized quality because its rate-distortion (R-D) model is not updated in real-time as time goes on.

In this work, we propose an avatar mobility and network condition-aware 3D game over wireless networks as shown in Fig. 1. One of the unique features of the proposed system is that 3D rendering level, target bit budget for an image compression, and packet transmission vector are dynamically adjusted based on the real-time avatar mobility and time-varying wireless network states while guaranteeing the user-preferred networking cost constraints.

II. Proposed Avatar Mobility and Network Condition-aware 3D Game

The proposed system is designed to provide the avatar mobility-based interactive 3D game with the mobile user considering compressed rendered image quality and user-preferred networking cost. The overall system architecture is given in Fig. 2. The proposed system consists of a command handler, a control unit, a GPU renderer, a Motion JPEG (MJPEG) encoder, a packetizer, and a networking cost-aware packet distributor. The proposed system obtains feedback information such as the packet-loss rate (PLR), available bandwidth, delay, and data usage from a mobile user. Based on this information, the control unit determines the optimal parameters, including the 3D rendering level, the target bit budget for image compression, and the packet transmission vector (i.e., the number of transmitted packets through each physical path). The command handler interprets the user-command to change the virtual-world scenes. Accordingly, the GPU renderer generates a rendered frame at the target rendering level, the MJPEG encoder compresses the rendered frame at the target bit budget, and the packetizer generates packets from the compressed stream. Finally, the networking cost-aware packet distributer assigns packets to selected physical paths according to the packet transmission vector.

978-1-4673-6770-7/15/$31.00 ©2015 IEEE

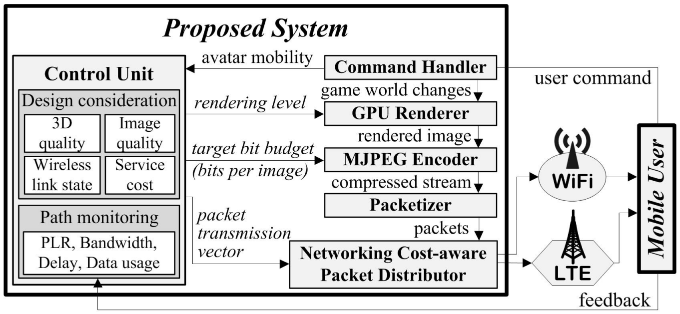


Fig. 2. Overall system architecture.

A. Problem Description

In general, 3D rendering quality is affected by various rendering levels, such as the view distance, realistic effects, and texture detail. The view distance, also knowns as rendering distance, is defined as the maximum distance between the avatar (representing a user in the virtual world) and objects that are drawn by the rendering engine. Thus, any objects beyond this distance are not rendered. In [5], the 3D rendering quality impairment function according to the view distance is derived using linear regression analysis. A rendered frame is compressed by the MJPEG encoder and streamed to the mobile user through wireless networks. The packet transmission vector for a compressed rendered frame is defined by $\overrightarrow{pt} = \left(pt_1, pt_2, ..., pt_{N_{net}}\right)$, where pt_k represents the number of transmitted packets through the k_{th} physical path for $1 \le k \le N_{net}$, and N_{net} is the number of available physical paths. The networking cost vector is symbolized by $\overrightarrow{nc} = \left(nc_1, nc_2, ..., nc_{N_{net}}\right)$, where nc_k denotes the required networking cost to transmit a packet through the k_{th} physical path for $1 \le k \le N_{net}$. Now, the total networking cost for transmitting a compressed rendered frame is defined by $\overrightarrow{pt} \bullet \overrightarrow{nc}$, where $\bullet$ denotes the inner product of two vectors. The target bit budget $\left(r_{tot}\right)$ of MJPEG encoder is calculated by

$$r_{tot}\left(\overrightarrow{pt}\right) = \sum_{k=1}^{N_{net}} pt_k \cdot S_{pt}, \tag{1}$$

where S_{pt} is the size of the packet payload.

Most state-of-the-art smart devices (such as Apple's iPhone series and Samsung's Galaxy series) have a restricted number of network interfaces due to the limited battery life and cost, etc. Besides, a network interface is generally connected to only one access network at a time. Thus, in real environments, N_{net} is set to 2 when a mobile device has a LTE modem and a WiFi interface. Now, we can formulate the problem as follows under the assumption that a free WiFi network is available.

Problem Formulation: Determine the view distance $\left(vd_i\right)$ and the number of transmitted packets through the LTE network $\left(pt_i^{LTE}\right)$ for the i_{th} frame with the determined pt_i^{WiFi} to minimize the following rendered and compressed image quality function

$$d\left(vd_i,\ r_i^{tot}\left(pt_i^{LTE}\right)\right)\Big|_{\text{with given } pt_i^{WiFi}}, \tag{2}$$

$$\text{subject to} \sum_{j=1+blk_i \cdot N_{frm}^{blk}}^{i} pt_j^{LTE} \cdot nc_{LTE} \le nc_{blk_i}^{avail}, \tag{3}$$

where pt_i^{WiFi} denotes the number of transmitted packets through the WiFi network, r_i^{tot} is the target bit budget for the i_{th} frame, N_{frm}^{blk} represents the number of frames per block, blk_i is the block index of the i_{th} frame defined by $blk_i = \left\lfloor (i-1)/N_{frm}^{blk} \right\rfloor$, nc_{LTE} represents the required networking cost to transmit a packet through the LTE network, and $nc_{blk_i}^{avail}$ denotes the available networking cost in the block blk_i calculated by

$$nc_{blk_i}^{avail} = \left(blk_i + 1\right) \cdot NC_{blk} - \sum_{j=1}^{blk_i \cdot N_{frm}^{blk}} pt_j^{LTE} \cdot nc_{LTE}, \tag{4}$$

where NC_{blk} represents the user-preferred networking cost per block. By the way, it is very difficult to mathematically describe the rendered and compressed image quality $d\left(vd_i,\ r_i^{tot}\left(pt_i^{LTE}\right)\right)\Big|_{\text{with given } pt_i^{WiFi}}$ since it includes both 3D rendering quality factor and compressed image quality factor. Hence, for the simplicity, the view distance is determined based on avatar mobility in Section II-B, and then the compressed image quality is controlled in Section II-D, separately.

B. Avatar Mobility-based View Distance Decision Algorithm

In general, when the avatar moves quickly, it would be better for the 3D game player to expand the avatar's view distance. In contrast, if the avatar moves slowly or is staying, a detailed representation of nearby objects around the avatar can improve the human visual perceptual quality at the cost of a short view distance. When reflecting on these characteristics, it is reasonable to adjust the view distance adaptively according to the avatar mobility. To avoid the abrupt view distance change that may cause serious annoying effect, the avatar mobility is low-pass filtered with the autoregressive-moving average (ARMA) model as follows.

$$\bar{v}_i = \omega_v \cdot v_i + \left(1 - \omega_v\right) \cdot \bar{v}_{i-1}, \tag{5}$$

where $\bar{v}_i$ denotes the average speed of avatar until the i_{th} frame, v_i is the instantaneous speed of avatar at the i_{th} frame, and ω_v is the weighting factor. Based on the low-pass filtered avatar mobility, the view distance is determined by

$$vd_{i+1} = \left(VD_{\max} - VD_{\min}\right) \cdot \frac{\bar{v}_i}{V_{\max}} + VD_{\min}, \tag{6}$$

where $VD_{\max}$ and $VD_{\min}$ denote the upper and lower bounds of the view distance values respectively, $V_{\max}$ represents the maximum speed.

C. Rate-Distortion Model Update Process

R-D modeling is essential technology for real-time image and video compression. In this work, an R-D model with a given view distance is estimated as follows.

$$\tilde{d}\left(r_{tot}\left(\overrightarrow{pt}\right)\right)\Big|_{\text{with given } vd} = \gamma \cdot e^{\varepsilon \cdot r_{tot}\left(\overrightarrow{pt}\right)}, \tag{7}$$

where $\gamma\ \left(\gamma \in \Re^{+}\right)$ and $\varepsilon\ \left(\varepsilon \in [-1,0]\right)$ are the model parameters. In fact, the R-D model should be continuously updated since it depends on the content included in each frame. However, when a variety of view distances are supported, the computational complexity required for R-D modeling of 3D rendered frames increases. To reduce the computational complexity, an R-D model of a rendered frame with a given view distance is interpolated from those of rendered frames with adjacent view distances. The reference R-D models are updated only when a scene change is detected, and the update range is limited to only a few view distances around the current view distance.

D. Lagrange Multiplier-based Optimization

The problem formulation in Section II-A is simplified as follows under the assumption that view distance is decided based on the avatar mobility.

Simplified Problem Formulation: Determine the number of transmitted packets through the LTE network $\left(pt_i^{LTE}\right)$ when vd_i and pt_i^{WiFi} for the i_{th} frame are given to minimize

$$\tilde{d}\left(r_i^{tot}\left(pt_i^{LTE}\right)\right)\Big|_{\text{with given } vd_i,\, pt_i^{WiFi}}, \tag{8}$$

$$\text{subject to} \sum_{j=1+blk_i \cdot N_{frm}^{blk}}^{i} pt_j^{LTE} \cdot nc_{LTE} \le nc_{blk_i}^{avail}. \tag{9}$$

Now, we describe how the optimal parameter pt_i^{LTE} is determined. Under the assumption that the WiFi bandwidth is fully utilized, pt_i^{WiFi} is determined by

$$pt_i^{WiFi} = \left\lfloor \frac{bw_{WiFi}}{FR \cdot S_{pt}} \right\rfloor, \text{ for } N_{frm}^{blk} \cdot blk_i < i \le N_{frm}^{blk} \cdot \left(blk_i + 1\right), \tag{10}$$

where FR is the frame rate, and bw_{WiFi} represents the available bandwidth through the WiFi network. To obtain the optimal solution in the given problem, Lagrange multiplier method is employed. Thus, the following penalty function is defined by combining the cost function Eq. 8 with the constraint Eq. 9.

$$PF\left(pt_i^{LTE}\right)\Big|_{\text{with given } vd_i,\, pt_i^{WiFi}} = \tilde{d}\left(r_i^{tot}\left(pt_i^{LTE}\right)\right) + \lambda \cdot \left(\max\left\{\sum_{j=1+blk_i \cdot N_{frm}^{blk}}^{i} pt_j^{LTE} \cdot nc_{LTE} - \frac{nc_{blk_i}^{avail}}{N_{frm}^{blk}} \cdot \left(i - blk_i \cdot N_{frm}^{blk}\right),\ 0\right\}\right), \tag{11}$$

where λ is the Lagrange multiplier. Since the penalty function in Eq. 11 is continuous and convex, a gradient method can find the optimal solution easily.

TABLE I. SYSTEM PARAMETERS.

Parameter	Value	Parameter	Value	Parameter	Value
VD_{max}	200 m	NC_{blk}	6 KRW	nc_{LTE}	0.018 KRW
VD_{min}	20 m	N_{frm}^{blk}	15	ω_v	0.01

III. EXPERIMENTAL RESULTS

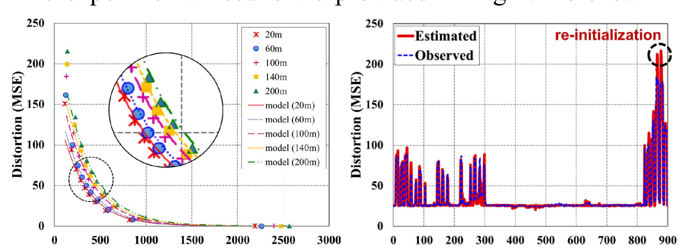
The proposed system is evaluated with VirtualGL [6] (open-source remote display software for 3D OpenGL applications) and PlaneShift [7] (online open-source 3D game software) using a mobile device connected to LTE and WiFi networks. Multipath TCP (MPTCP) [8] is adopted as the transport protocol of VirtualGL to transmit packets through multiple physical paths, and "libjpeg-turbo" library [9] is used for JPEG compression and decompression. In PlaneShift, the screen resolution is fixed at 640×480 pixels, and the frame rate (FR) is set to 15 frames per second. The networking cost model for the LTE network is determined based on a commercial service model (SKT LTE DATA 130) provided by SK-Telecom, the largest South Korean wireless telecommunications operator. The experiment is performed over 60 s (900 frames). The other system parameters are summarized in Table I.

A. Performance Verification of the Proposed R-D Model

In this section, we verify the accuracy of the proposed R-D model. At the initialization step, we generate 10 R-D models (from distance VD_{min} to VD_{max} in 20 m intervals). Each model includes 10 sampling points. Figure 3 (a) shows an example of the initialized R-D model (models for distances 40 m, 80 m, 120 m, 160 m, and 180 m are omitted from this figure due to space constraints). Figure 3 (b) represents the observed and estimated distortion per frame. If a scene change is detected (i.e., when the distortion error is higher than the threshold value), the R-D models are re-initialized. On average, the R-D model achieves an estimation-error rate of approximately 5.49%. Consequently, the proposed model fits the observed data well.

B. Performance Comparison with Existing Algorithms

Now, the avatar mobility-based view distance adaptation algorithm is examined. In PlaneShift, the avatar's mobility can be controlled in three patterns—stop, walk, and run—and the velocity of each is 0 m/s, 1.4 m/s, and 5.6 m/s, respectively. The experimental results are provided in Fig 4. As shown in



(a) (b)

Fig. 3. Performance verification of the proposed R-D model: (a) observed and estimated output bits per frame, and (b) observed and estimated distortion.

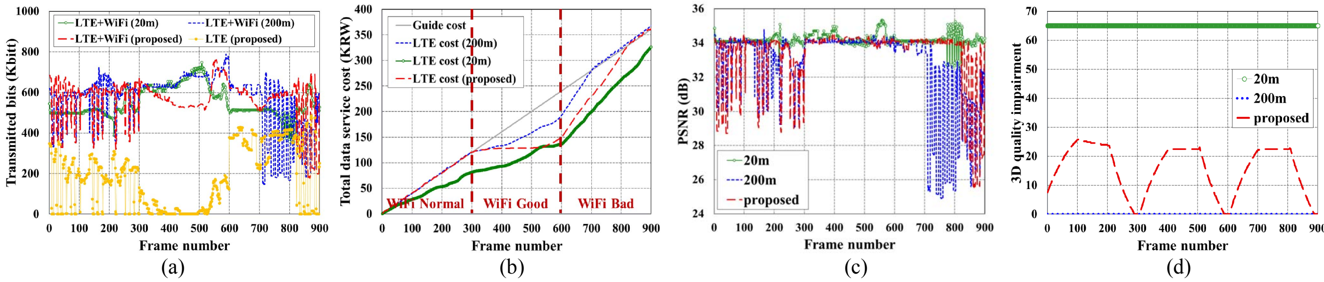


Fig. 5. Performance comparison with static view distance-based algorithms: (a) WiFi channel states, (b) amount of transmitted bits, (c) PSNR, and (d) 3D rendering quality impairment.

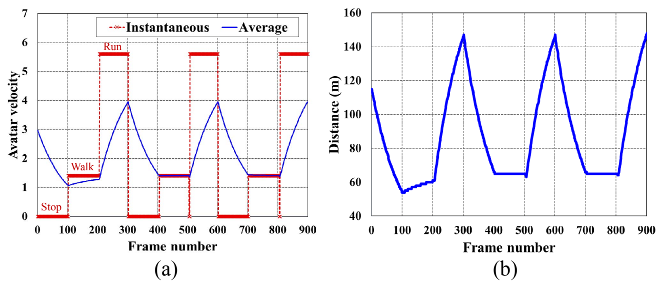


Fig. 4. Performance verification of the proposed avatar mobility-based view distance adaptation algorithm: (a) avatar velocity, and (b) view distance.

figure, the view distance is adjusted dynamically in the range of $VD_{\min}$ to $VD_{\max}$, according to the avatar's mobility. It is apparent that the view distance is completely synchronized with the average velocity of the avatar.

In the following, the performance of the proposed rendered and compressed image quality optimization algorithm is compared with existing static view distance-based algorithms. We design a 3D rendering quality-oriented algorithm that is adopted to maximize the perceived 3D rendering quality by setting the view distance to $VD_{\max}$. In addition, an image quality-oriented algorithm is implemented to improve PSNR by setting the view distance to $VD_{\min}$. The experimental results are provided in Fig 5. As shown in figures, in the 3D rendering quality-oriented algorithm, 3D rendering quality significantly improved with a longer view distance. However, the image quality is not sufficiently guaranteed after the 700th frame. On the other hand, the image quality-oriented algorithm achieved a higher PSNR than other algorithms. However, 3D rendering quality can be seriously degraded because of an extremely low view distance. In the proposed algorithm, the quantity of bits transmitted through the LTE network is dynamically adjusted considering WiFi channel states and user-preferred networking cost to maximize rendered and compressed image quality, as shown in Fig. 5 (a) and (b). As a result, the proposed algorithm efficiently guarantees the relatively high PSNR while achieving a balance in 3D rendering quality, as shown in Fig. 5 (c) and (d). The experimental results are summarized in Table II.

TABLE II. PERFORMANCE COMPARISON WITH 3D RENDERING QUALITY-ORIENTED AND IMAGE QUALITY-ORIENTED ALGORITHMS.

	Image quality-oriented (20 m)	**3D rendering quality-oriented (200 m)**	**Proposed (adaptive)**
View distance (m)	20	200	89.31
3D rendering quality impairment	67.13	0	13.92
Amount of LTE transmitted bits (Kbit)	165.10	179.29	22.50
PSNR (dB)	34.19	32.45	33.16

IV. CONCLUSION

In this work, we have presented an avatar mobility and network condition-aware 3D game over wireless networks. The proposed system determines the view distance, target bit budget, and packet transmission vector in an R-D optimized way according to the avatar's mobility and time-varying wireless network states. The experimental results have demonstrated that the proposed system guarantees the high image quality while maintaining a balance with 3D rendering quality in a cost effective way.

ACKNOWLEDGMENT

This work was partly supported by the ICT R&D program of MSIP/IITP [13-911-04-005, Research and Development of 5G Mobile Communications Technologies using CCN-based Multi-dimensional Scalability] and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2006732).

REFERENCES

[1] IDC, "Worldwide semiannual public cloud services tracker," June 2014.

[2] Juniper Research, "Mobile cloud: smart device strategies for enterprise & consumer markets 2011–2016," July 2011.

[3] R. Shea, J. Liu, E.-H. Ngai, and Y. Cui, "Cloud gaming: architecture and performance," IEEE Network, vol. 27, no. 4, pp. 16–21, 2013.

[4] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," IEEE Transactions on Multimedia, vol. 15, no. 4, pp. 870–883, 2013.

[5] L. Yao, S. Wang, and S. Dey, "Modeling, characterizing, and enhancing user experience in cloud mobile rendering," IEEE Computing, Networking and Communications (ICNC), 2012.

[6] VirtualGL. Available: http://virtualgl.org/

[7] PlaneShift. Available: http://www.planeshift.it/

[8] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural guidelines for multipath TCP development," IETF RFC 6182, March 2011.

[9] Libjpeg-turbo Library. Available: http://www.libjpeg-turbo.org/