A Joint Asymmetric Graphics Rendering and Video Encoding Approach for Optimizing Cloud Mobile 3D Display Gaming User Experience

Yao Lu, *Student Member, IEEE*, Yao Liu, *Member, IEEE*, Sujit Dey, *Fellow, IEEE*

Abstract—**With the development and deployment of ubiquitous wireless network together with the growing popularity of mobile auto-stereoscopic 3D displays, more and more applications have been developed to enable rich 3D mobile multimedia experiences, including 3D display gaming. Simultaneously, with the emergence of cloud computing, more mobile applications are being developed to take advantage of the elastic cloud resources. In this paper, we explore the possibility of developing Cloud Mobile 3D Display Gaming, where the 3D video rendering and encoding are performed on cloud servers, with the resulting 3D video streamed to mobile devices with 3D displays through wireless network. However, with the significantly higher bitrate requirement for 3D videos, ensuring user experience may be a challenge considering the bandwidth constraints of mobile networks. In order to address this challenge, different techniques have been proposed including asymmetric graphics rendering and asymmetric video encoding. In this paper, for the first time, we propose a joint asymmetric graphics rendering and video encoding approach, where both the encoding quality and rendering richness of left view and right view are asymmetric, to enhance the user experience of the cloud mobile 3D display gaming system. Specifically, we first conduct extensive user studies to develop a user experience model that takes into account both video encoding impairment and graphics rendering impairment. We also develop a model to relate the bitrate of the resulting video with the video encoding settings and graphics rendering settings. Finally we propose an optimization algorithm that can automatically choose the video encoding settings and graphics rendering settings for left view and right view to ensure the best user experience given the network conditions. Experiments conducted using real 4G-LTE network profiles on commercial cloud service demonstrate the improvement in user experience when the proposed optimization algorithm is applied.**

Index Terms— **Cloud Mobile Gaming, 3D, user experience**

I. Introduction

The growing popularity of auto-stereoscopic 3D displays for mobile devices, together with ubiquitous wireless networks, have fueled an increasing user expectation for rich 3D mobile multimedia experiences, including 3D display gaming. According to [1], the world's first glass free 3D tablet was released in December 2013; and by August 2014, there are already more than 10 different brands of glass free 3D tablets in the market. Recently, a new architecture called Cloud Mobile 3D Display Gaming (CMG(3D)) has been proposed to leverage the growing trend of glass free 3D mobile devices, and bring true 3D gaming experience to mobile users. Figure 1 shows the CMG(3D) architecture [4], where the 3D rendering is performed on the cloud server in response to gaming commands from the mobile device, instead of the mobile device itself. This architecture extends mobile cloud gaming for 2D devices that were introduced earlier [6]. For real 3D gaming experience, we place two virtual cameras in the game world to generate a left view and a right view of the game video. After the two views are generated, they will be encoded as a 3D video and transmitted through wireless network to the mobile device, and displayed on the device 3D screen. On the reverse side, game commands are transmitted from mobile device to the game server through wireless network. In this way, users can play 3D games as if the game is rendered locally, but with the advantages of a thin client, and no need to download/store each game to the mobile device. Moreover, the game developers can develop a single version of the game that runs on the cloud servers instead of having to develop platform/device specific versions.

Although the CMG(3D) architecture has great advantages compared to the traditional 3D gaming architecture, the challenge shifts from client side to server side on how to stream high quality 3D video through fluctuating network conditions with low latency.

To address the above problem, asymmetric video encoding techniques, which dynamically set QP/resolution differently for two views, can be potentially used to lower bitrate while minimizing impact on user experience [2][3]. More recently, asymmetric graphics rendering, which dynamically sets rendering parameters like texture detail and/or view distance differently for two views, has been proposed to lower bitrate while attempting to preserve user experience [4]. However, the above techniques are studied and applied separately. In this paper, we explore the possibility of combining asymmetric

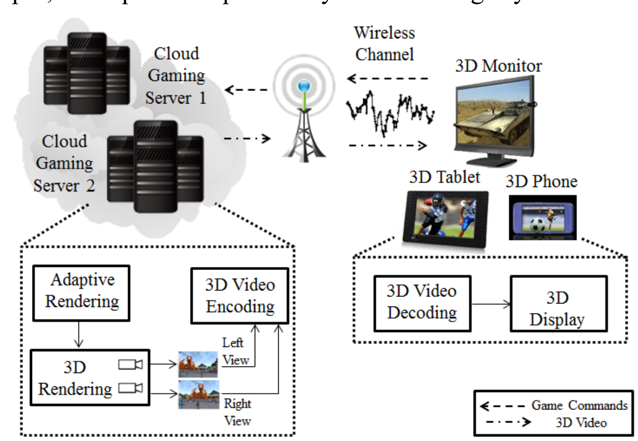

Figure 1. Architecture of Cloud based Mobile 3D Display Gaming System

video encoding and asymmetric graphics rendering techniques together to jointly optimize the whole system and thus to provide the best user experience. To quantitatively measure the user experience by different video encoding settings and graphics rendering settings, we perform subjective tests and develop a user experience model. Moreover, we also develop a rate model to estimate the bitrate value as a function of video encoding settings and graphics rendering settings. By making use of the above two models, we further propose an optimization algorithm to automatically choose the video encoding settings and graphics rendering settings for the left and right views separately to ensure the best user experience given the network conditions.

The rest of the paper is organized as following: Section II reviews related work. In Section III, we propose a user experience model to quantitatively measure the user experience according to different video encoding settings and graphics rendering settings. In Section IV, we propose another model to estimate the video bitrate as a function of video encoding settings and graphics rendering settings. Section V proposes an adaptation algorithm to automatically choose the best video encoding settings and graphics rendering settings given certain mobile network conditions. Section VI shows the experimental results using our CMG(3D) prototype hosted on Amazon Web Service and using commercial cellular network profiles. Section VII proposes future work and concludes the paper.

II. Related Work

As introduced in Section I, our objective is to reduce the bitrate and thereby reduce latency of the 3D video transmitted from the CMG(3D) cloud server to the 3D mobile device depending on the wireless network bandwidth available, while also preserving high perceived video quality. Several techniques have been proposed to optimize video encoder or rendering engine for the above purpose. Asymmetric 3D video encoding [2][3], where the videos of the left and right views are encoded with different quality, can be potentially used to reduce the 3D video bit rate, and hence the delay, while transmitting over constrained wireless networks. Recently, another technique called asymmetric graphics rendering has been introduced [4] in which the rendering engine chooses different texture detail or view distance settings for left view and right view in a way that reduces the bit rate of the resulting encoded video in order to meet network constraint. However, none of the above work has considered the possibility of optimizing the whole system by applying both asymmetric video encoding and asymmetric graphics rendering at the same time. In addition, because we want to apply these two techniques at the same time, we need to first understand how the joint effects influence user experience. In the literature, [5] propose a user experience model but it only considers video encoding impairment and it is based on a 2D display CMG system. Authors of [6] have done user experience study of considering both video encoding impairment and graphics rendering impairment but the above is applicable only for single view video for 2D displays. Authors of [4] have done user study based on CMG(3D) system but it does not include video encoding impairment. Thus, in this paper, for the first time, we study the user experience considering the joint effect of video encoding impairment and graphics rendering impairment. To optimize jointly the video rendering and encoding settings for a given network constraint, we also need to develop a model to estimate the resulting video bitrate using given video rendering and encoding settings. Different from [7] which only include video encoding settings, our model also takes into account the influence of graphics rendering settings.

III. User Experience Model

In this section, we study and model user experience considering the joint effect of video encoding impairment and graphics rendering impairment, as opposed to [4] which takes into account the effects of graphics rendering impairment only. We use the same Cloud Mobile 3D Display Gaming Mean Opinion Score (CMG(3D)-MOS) introduced in [4] as a measurement metric for modeling CMG(3D)-UE, as shown in equations (1) and (2):

$$\text{CMG(3D)-MOS} = 1 + 0.035R + 7\times10^{-6}R\left(R-60\right)\left(100-R\right) \quad (1)$$

$$R = 100 - I\left(VE, GR\right) \quad (2)$$

In Equation (1), the CMG(3D)-MOS metric is formulated by a transmission rating factor R, which represents the overall user experience. R factor takes value between 0 and 100; higher R value corresponds to higher CMG(3D)-MOS and better user experience. In Equation (2), the term I stands for the combined impairment caused by video encoding (VE) and graphics rendering (GR).

In the following, we describe how we conducted subjective tests to derive the impairment functions and how we validated them.

A. *Subjective Test settings*

Table I shows the specific video encoding settings and graphics rendering settings we want to include in our experiment. In detail, for video encoding settings, we fix the resolution to be VGA (640x480) and frame rate to be 25fps in our experiment but change the quantization parameter (QP) only. For graphics rendering, we study the effect of asymmetric texture detail, but not the view distance as our subjective experiment shows that by applying asymmetric view distance, users will feel dizzy after playing the game for a long time. Texture detail defines the quality of the images on the surface of the objects. As is defined in [4], we define texture detail to be high when the game is using the original texture images, to be medium when the texture images are downsampled once and low when the texture images are downsampled twice.

Figure 2 shows the testbed used for the subjective tests. We use a 3D monitor with a laptop to substitute for 3D display of mobile devices because current available mobile 3D displays do not have as good quality as 3D monitors that may cause additional impairment which we want to avoid. The laptop is connected to a network emulator via an access point and the network emulator is connected to the game server. The selected game which runs upon the above framework is an online open-source MMORPG game PlaneShift [8]. We then invited 15 students (10 male, 5 female; aged 18~24) to participate in our subjective experiments. Firstly, we asked the testers to sit before a 23 inch LG D2342 3D Monitor, and show them a 3D video as a training sequence before the real test starts to let the testers adjust their viewing angle. After that, we start the game and manually set the video encoding settings and graphics rendering settings according to Table I independently for each view. Once a combination of rendering and encoding factors is set, we ask the testers to play the game for 1 minute and evaluate the impairment according to the criterion listed in Table II at the end of each condition. During the whole experiment, the testers were asked to control the avatar to perform multiple tasks (including attacking an enemy, looking for an object, talking to an NPC, etc.).

Figure 2. Testbed for Subjective Experiments

TABLE I. EXPERIMENT SETTING

Settings	***Experiment Values***
Texture Detail(Down Sample)	High(0) Medium(2) Low(4)
Quantization Parameter	25 27 29 31 33 35 37 39

TABLE II. IMPAIRMENT CRITERION

I	***Description***
0	No visual impairment
0-20	Minor visual impairment
20-40	Noticeable visual impairment
40-60	Clear visual impairment
60-100	Unacceptable visual impairment

TABLE III. QP THRESHOLD FOR DIFFERENT TEXTURE DETAIL SETTINGS

Texture Detail	H	M	L
I_{TD}	29	31	35

TABLE IV. AVERAGE I_{TD} SCORES FOR DIFFERENT TEXTURE DETAIL COMBINATIONS

Texture Detail Combination	H-H	H-M	M-M	M-L	L-L
I_{TD}	0.2	8.35	12.3	19.14	24.5

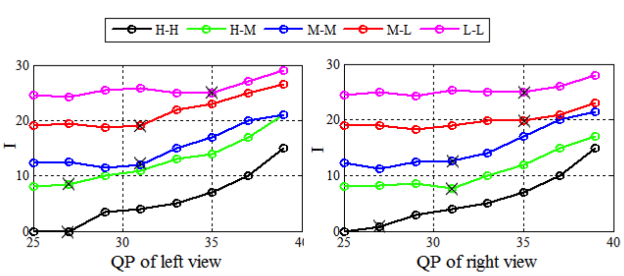

Figure 3. (a) left, Relationship between I and QP_L under different texture detail combinations (b) right, Relationship between I and QP_R under different texture detail combinations

B. Impairment Function Derivation

Considering the graphics rendering and video encoding settings used in our CMG(3D) platform and the two views, we can set the impairment function as follows.

$$I = I(VE, GR) = I(TD_L, TD_R, QP_L, QP_R) \tag{3}$$

where TD means texture detail, QP means quantization parameter. The subscripts L and R represent left view and right view respectively. In order to study I, we first keep one of the four parameters fixed to its best quality value during the test and see how the impairment I changes according to other three parameters.

Figures 3(a) and 3(b) show the results (average impairment values) when we keep QP of one of the views to be 25 (right view for (a) and left view for (b)) but change QP of the other view and at the same time change the texture detail settings for both views. From these two figures we can clearly observe that for each texture detail combination, the impairment values remain similar till some QP level (marked as cross in the figure), beyond which the impairment increases. For example, for curve L-L in Figure 3(a), the threshold is at QP = 35. Clearly the value of this threshold is related to the texture detail setting. Further, we found that this threshold was only related to the texture detail setting of the view whose QP is changing. For example, in the Figure 3(a), the threshold of M-M is the same as the threshold of M-L and that value is also very close to the threshold of H-M and M-M in Figure 3(b). Thus, it means for a specific texture detail setting of one view, there is a corresponding QP value so that when QP is less than or equal to that value, the video encoding won't cause additional impairment besides the impairment caused by texture detail. According to the results from Figures 3(a) and (b), we list the threshold and the corresponding impairment I_{TD} in Tables III and IV. Note that although the above relationship as well as all the models derived subsequently are general, the values of the parameters need to be derived for specific games. The values in this paper are derived based on the game Planeshift [8].

Because we have this threshold, we propose to model I by two parts. The first one is the I_{TD} that is the impairment caused by texture detail and the second part is I_A that is the additional impairment when QP is bigger than the threshold. Equation (4) shows the relationship.

$$I(TD_L, TD_R, QP_L, QP_R) - I_{TD}(TD_L, TD_R) = I_A$$
$$= \begin{cases} 0 & QP_L < T(TD_L),\ QP_R < T(TD_R) \\ f(QP_L - T(TD_L)) & QP_L \geq T(TD_L),\ QP_R < T(TD_R) \\ f(QP_R - T(TD_R)) & QP_L < T(TD_L),\ QP_R \geq T(TD_R) \\ g(QP_L - T(TD_L), QP_R - T(TD_R)) & QP_L \geq T(TD_L),\ QP_R \geq T(TD_R) \end{cases} \tag{4}$$

in which I_{TD} is the value form Table IV. T function is the value from Table III.

Next, we model $f(x)$ shown in Equation (5); we use nonlinear regression to derive this relationship and coefficients as shown in Figure 4. The MSE of the regression is 0.8491.

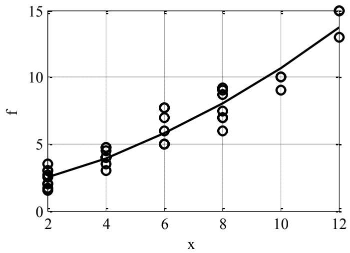

Figure 4. Derivation of f(x)

$$f(x) = ax^2 + bx + c \qquad (5)$$

in which a=0.05, b=0.41, c=1.49 for the game Planeshift.

For the g(x_1, x_2) function in Equation (4), a multivariable nonlinear regression is used to derive the relationship and coefficients shown in Equation (6). The MSE of the regression is 0.2366.

$$g(x_1, x_2) = ax_1^2 + ax_2^2 + bx_1 + bx_2 + cx_1x_2 + d \qquad (6)$$

in which a=0.043, b=0.7, c=-0.027, d=0.978 for the game Planeshift.

Equations (3) (4) (5) and (6) complete our user experience model. We will validate it in the next subsection.

C. *Model Validation*

In order to validate the impairment model (Equations (3), (4), (5) and (6)) derived in the previous subsection, we conducted another set of experiments with a new group of 16 participants (10 male, 6 female; aged 18~25), playing the same game. This time, they give the CMG(3D)-UE score (Equation (1)) varying between 1.0 to 4.5 according to Table V. In addition, in this set of experiments, the texture detail and quantization parameters for both views are changed at the same time. Furthermore, to ensure the generality of the model, the subjects are asked to evaluate three different scenes. Figures 5(a)(b)(c) show the relationship between predicted CMG(3D)-UE score computed by the derived impairment function (y-axis) and subjective CMG(3D)-UE score given by human subjects (x-axis) for the three scenes. In the figures, each data point represents one combination of graphics rendering settings and video encoding settings. We also plotted 95% confidence interval for each measurement as blue lines in the figure to show the variety among different subjects. The correlation in scene 1 is 0.96 while it is 0.95 for scene 2 and 0.97 for scene 3. The above results show the accuracy of the derived CMG(3D)-UE model, and its applicability to different scenes.

TABLE V. CMG(3D)-UE CRITERION

CMG(3D)-MOS	*Description*
4.5	No visual impairment
4.0-4.5	Minor visual impairment
3.0-4.0	Noticeable visual impairment
2.0-3.0	Clear visual impairment
1.0-2.0	Unacceptable visual impairment

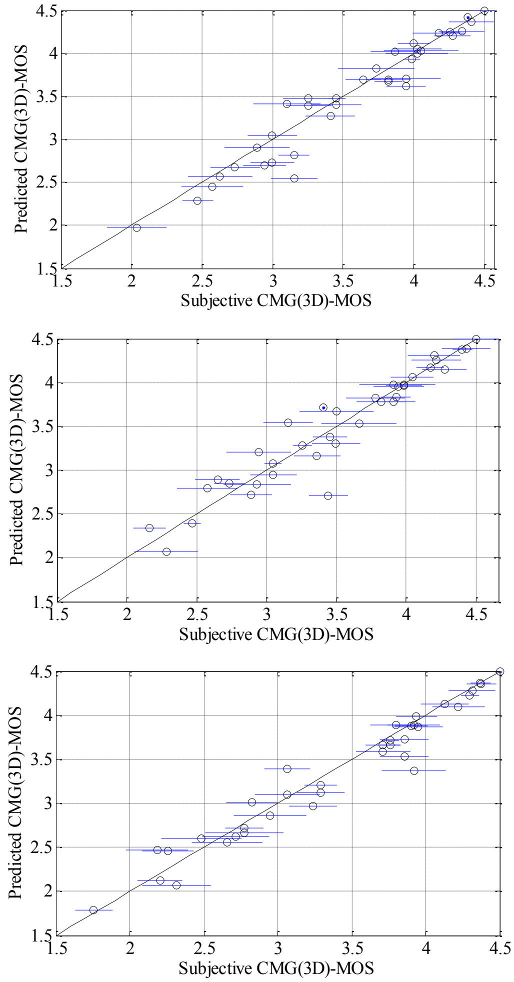

Figure 5. Validation of CMG(3D)-MOS with blue line showing 95% confidence interval (a) top, results for scene 1 (b), middle results for scene 2 (c) bottom, results for scene 3

IV. BITRATE MODEL

In this section, we develop a bitrate model that estimates the encoding bitrate of a view from both the graphics rendering and video encoding settings for this view.

Several techniques have been proposed to model the bitrate of encoded video as a function of the video encoding parameters. For example, in [8], Ma et al. proposed Equation (7) to model bit rate R using quantization step q and video frame rate t.

$$R(q,t) = R_{\max}\left(\frac{q}{q_{\min}}\right)^{-a}\left(\frac{t}{t_{\max}}\right)^{b} \qquad (7)$$

In Equation (7), coefficients q_{min} and t_{max} represent the minimum quantization step and the maximum frame rate, respectively, and are chosen based on the application; R_{max} indicates the maximum bitrate when encoding a video at q_{min} and t_{max}; coefficients a and b are model parameters that depend on the content of the video. The authors in [7] further proposed a method to estimate a and b based on content features shown in Equation (8) and (9).

$$[a \;\; b]^T = B[1 \;\; \mu_{FD} \;\; \mu_{MVM} \;\; \frac{\mu_{MVM}}{\sigma_{MDA}}]^T \qquad (8)$$

$$B = \begin{bmatrix} 1.1406 & -0.0330 & -0.0611 & 0.1408 \\ 0.4462 & 0.0112 & 0.0680 & -0.0667 \end{bmatrix} \quad (9)$$

in which μ_{FD} represents mean of frame difference, μ_{MVM} stands for mean of motion vector magnitude and σ_{MDA} means standard deviation of motion direction activity.

Though reported in [7] that this model is of high accuracy, it is based on a data set containing videos that are natural scene videos and the resolutions are CIF (352x288) rather than the situation in our CMG(3D) application where the view is generated by computer instead of camera in the real world and the video resolution for each view is VGA (640x480). Moreover, this work only considers video encoding parameters but do not consider any graphics rendering settings. Thus, the model equation (7) ~ (9), especially the model parameter may not be accurate enough in our CMG(3D) application. Therefore, in this paper we extend their work by:

1. Performing experiments using CMG(3D) videos with VGA resolution to validate the model equations.
2. Performing additional experiments to adjust the model of parameter *a* (Equation (8)(9)) by incorporating graphics rendering setting.

A. Model Equation Validation

In this subsection, we introduce how we perform experiments to validate the model equations. Firstly, because in this paper, we do not evaluate the influence of framerate to user experience or bitrate, we will fix the framerate and set t to be t_{max} and thus we can simplify the equation as:

$$R(q) = R_{\max}\left(\frac{q}{q_{\min}}\right)^{-a} \quad (10)$$

In order to validate this model, we captured 3 videos using our CMG(3D) system with different texture detail settings and encode them using different QP settings from Table I. Using the H.264/AVC standard definition that q=2^((QP-4)/6), the corresponding q values are 11, 14, 18, 23, 28, 36, and 45. For each video, we encode it by using x264 encoding library and record the bitrate under each q value. We set R_{max} to be the bitrate when encoding with q_{min} and calculate normalized bitrate $R(q)/R_{max}$. Figure 6 shows the results. X-axis of the figure is q which ranges from 11 to 45 and y-axis is the normalized bitrate. The results of each video are represented by a specific color. Besides the bit rates shown as circles for the 3 videos with different texture details, we also plot a line for each video to represent the model equation. The parameter a is obtained by minimizing the squared error between the model predicted and measured rates for each video. From figure 6, we can conclude that Equation (10) can model the bit rate of CMG(3D) videos with high accuracy.

B. Model Parameter Prediction

In this subsection, we discuss how we adjust the parameter model (Equation (8)(9)) proposed in [7] to cope with CMG(3D) application. As is reported in [7] that μ_{FD}, μ_{MVM} and μ_{MVM}/σ_{MDA} are the most related content features which influence parameter a. However, as is shown in Figure 6, the parameter a will vary for different texture detail settings. Thus we combine texture detail, TD, with the content features in Equation (8) as the input parameters for predicting a. Further, we captured 18 video clips with different texture detail settings and in different scenes performing different tasks and we use the same generalized linear predictor with leave-one-out cross-validation error method reported in [7] to derive the equations, which are shown in Equation (11)(12).

$$a = B[1 \ \mu_{FD} \ \frac{\mu_{MVM}}{\sigma_{MDA}} \ TD]^T \quad (11)$$

$$B = \begin{bmatrix} 1.2 & -0.068 & 0.00017 & 0.0024 \end{bmatrix} \quad (12)$$

Figure 7 shows the validation of the proposed model parameter prediction (Equations (11) (12)). Our results show that using μ_{FD}, μ_{MVM}/σ_{MDA} and TD is sufficient to predict the model parameter accurately.

Thus, Equations (10) (11) (12) completes our bitrate model.

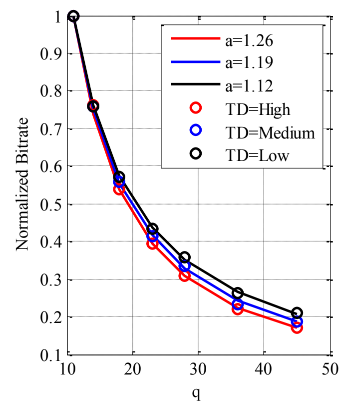

Figure 6. Validation of model equation

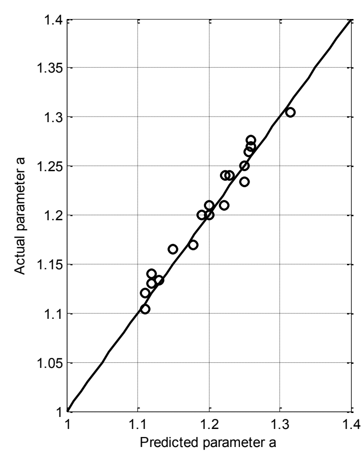

Figure 7 Validation of model parameter

Given:
1) Network bandwidth BW
2) Texture Detail bound TD_{min} and TD_{max}
3) Quantization Parameter bound QP_{min} and QP_{max}
4) Current content features μ_{FD}, μ_{MVM} and σ_{MDA}

Find:
The optimal TD_L, TD_R, QP_L, and QP_R to minimize impairment I

$$I^{opt} = \min(I)$$

s.t.

$$TD_{\min} \le TD_L \le TD_R \le TD_{\max}$$

$$QP_{\min} \le QP_L \le QP_R \le QP_{\max}$$

$$R_L(TD_L, QP_L) + R_R(TD_R, QP_R) \le BW$$

Figure 8. Problem Formulation

Algorithm Joint Asymmetric Video Encoding and Asymmetric Graphics Rendering Adaptation (*AVARA*)
Input: 1) Network bandwidth BW 2) Content features μ_{FD}, μ_{MVM} and σ_{MDA}
Output: I^{OPT}, TD_L^{OPT}, TD_R^{OPT}, QP_L^{OPT}, QP_R^{OPT}

Step 1: Initialize upper bound and lower bound of TD_L, TD_R, QP_L, QP_R; set iteration to be 0; set I^{OPT} to be infinite;
inqueue(*original_problem*, *subproblem_queue*)
Step 2: While (length(*subproblem_queue*) > 0 &&
iteration < MAX_INTERATION)
iteration = *iteration* +1
subproblem = outqueue(*subproblem_queue*)
Step 3: [TD_L, TD_R, QP_L, QP_R , I] = s =
non_linear_continuous_variable_optimization(I)
Step 4: If s = Integer solution
If $I < I^{OPT}$
$I^{OPT} = I$
[TD_L^{OPT}, TD_R^{OPT}, QP_L^{OPT}, QP_R^{OPT}] =
[TD_L, TD_R, QP_L, QP_R]
Endif
Else if s = Non-integer solution
If $I([TD_L], [TD_R], [QP_L], [QP_R]) < (1+m\%)I$
$I^{OPT} = I$
[TD_L^{OPT}, TD_R^{OPT}, QP_L^{OPT}, QP_R^{OPT}] =
[$[TD_L]$, $[TD_R]$, $[QP_L]$, $[QP_R]$]
continue
Endif
For i=1; i<$2^{\text{number\_of\_non\_integer\_values}}$; i++
Determine the i^th^ variable range for *subproblem$_i$*
inqueue(*subprobelm$_i$*, *subproblem_queue*)
Endfor
If s = No feasible solution
continue
Endif
Endwhile
Step 3: Return [TD_L^{OPT}, TD_R^{OPT}, QP_L^{OPT}, QP_R^{OPT} ,I^{OPT}]

Figure 9. Pseudo code of AVARA algorithm

V. Optimization Algorithm

In Section III and IV we have proposed 1) a user experience model which models cloud mobile 3D display gaming user experience as a function of video encoding settings and graphics rendering settings, and 2) a bitrate model which estimates video bitrate needed to encode the rendered video as a function of video encoding settings and graphics rendering settings. In this section, we combine these two models so that by selecting proper graphics rendering (texture detail) settings and video encoding (quantization parameter) settings of the rendered video, we can find an optimal solution for maximizing user experience given a network bandwidth limit.

Figure 8 shows the problem formulation. Because according to Equation (1) and (2), maximizing CMG(3D)-UE is equal to minimizing I, we set our optimization target as minimizing I. TD_{min}, TD_{max}, QP_{min} and QP_{max} are the minimum and maximum boundaries of the settings being used for a game.

In order to solve the above problem, we propose Joint Asymmetric Video Encoding and Asymmetric Graphics Rendering Adaptation (*AVARA*) algorithm which runs periodically (in this paper we set the period to be 1 second). At the beginning of each time period, we will obtain the inputs of this algorithm: 1) network bandwidth BW and 2) current content features (μ_{FD}, μ_{MVM} and σ_{MDA}). We use content features in the previous time period with Equation (16) to estimate the bitrate consumption in the current time period. The output of the algorithm will be the optimal video encoding settings and graphics rendering settings for the current time period.

Figure 9 shows the *AVARA* algorithm. As this problem is a discrete variable, non-linear objective function, with unequal non-linear constraint functions, the core idea in this algorithm is based on classic branch and bound algorithm [11]. Specifically, the original problem with an integer variable is first relaxed to an optimization problem with continuous variable (the first *subproblem*). After this, the program generated *subproblems* where the possible range of the variable (still continuous) is being reduced. Then it solves these *subproblems*. This process continues until the variable is fixed to a (integer) value. However, the disadvantage of the traditional branch and bound algorithm is that it is complex and time consuming while our application needs to meet real time computation constraint. Therefore, we add an "early exit" condition to signifiantly decrease the complexity while still achieving a near-optimal solution. The way we do it is that after every time we get a solution of the *subproblem* and if it is a non-integral solution, and before it is going to divide the current *subproblem* into k *subproblems*, we will round up the values of the variable and check whether this solution can get an I^{OPT} which is within m% error of the optimal non-integral solution. We have implemented the algorithm in C; with m=2, the average running time for each time period is 1.3 ms on an Intel Xeon E5-2670 @2.60GHz processor with 15GB memory, demonstrating real-time adaptation capability.

VI. Experimental Results

In this section, we report on experiments conducted using a commercial cloud service, Amazon Web Service (*AWS*) [12], to verify the performance improvement by applying the proposed Joint Asymmetric Video Encoding and Asymmetric Graphics Rendering Adaptation (*AVARA*) technique. We use the same testbed as shown in Figure 2, except that 1) we put a network simulator between AP and laptop and 2) we implement our CMG(3D) system, including the *AVARA* algorithm, on *AWS* servers. In detail, we modified the open-source game engine Planeshift and programmed a streaming software so that 1) the game engine is able to pre-load different levels of textures when initializing the game loop so that when one specific texture detail setting is chosen it is able to switch to it very fast. 2) the streaming software will encode the left and right view videos and stream the videos to the client through TCP, and the algorithm is also implemented there using C++ to decide the best TDs and QPs 3) the streaming software is hooked with the game engine and they share some variables through process level share memory mechanism and in this way the pixels of left view and right view can be shared from game engine to the streaming software and the decision of the texture detail setting computed by the algorithm from the streaming software can be shared to game engine so that game engine can set the texture detail in its game loop. For the Amazon cloud server, the CPU is Intel Xeon E5-2670 @2.60GHz with 15GB memory and the GPU has 1536 CUDA cores and 4GB of video memory. The operating system is Windows Server 2008 R2 SP1.

We firstly collected real 4G-LTE network traces by using network bandwidth testing software Speedtest.net [10] to record the bandwidth. Figure 10(a) shows a sample LTE trace, which is emulated using the network emulator in our testbed. We then measured the bandwidth from Amazon cloud server to our lab. We use iPerf to test, with testing every 10 minutes from 8:00 a.m. to 10:00 p.m. for three days. Figure 11 shows the PDF of the results. We find that the bandwidth has some variance but is quite adequate. Compared to Figure 10(a) where the largest

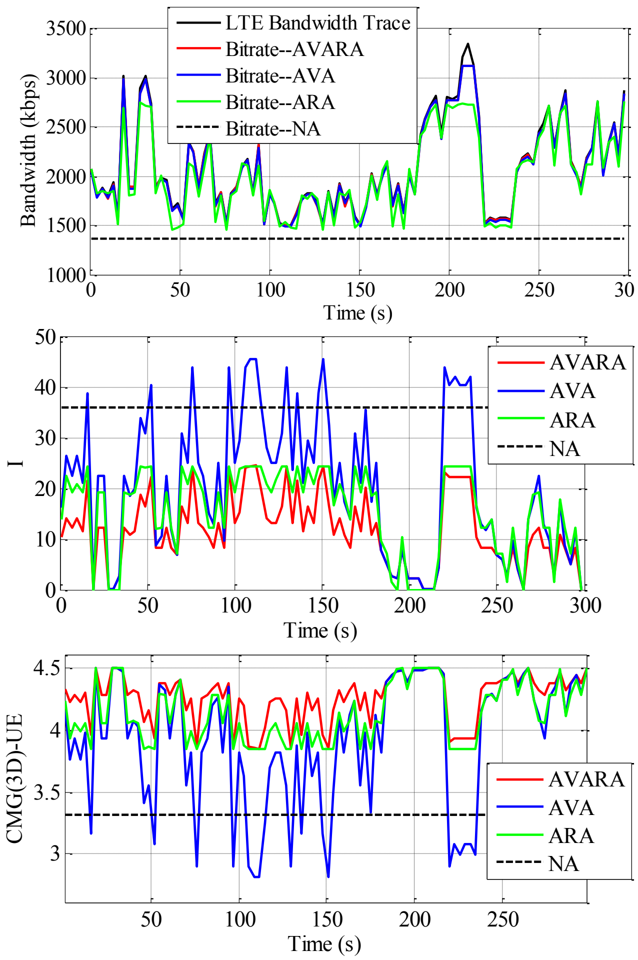

Figure 10. (a) top, LTE bandwidth trace and bandwidth consumption of the algorithms (b) middle, resulting impairment *I* of the algorithms (c) bottom, CMG(3D)-UE of the algorithms

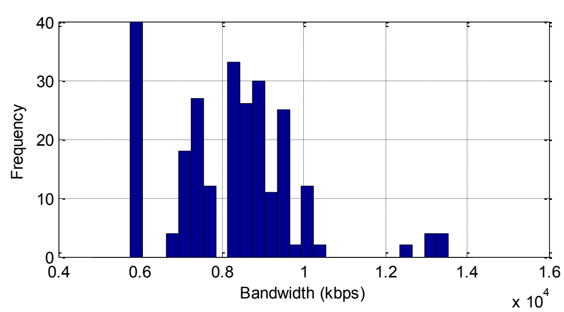

Figure 11. PDF of the bandwidth from Amazon cloud server to UCSD Mobile System Design Lab

bandwidth of LTE is about 3.4Mbps, the lowest bandwidth from AWS to our lab is about 5.8Mbps. Thus we conclude the main bandwidth bottleneck is caused by the LTE trace. In addition, for comparison purpose, we also implemented three other algorithms. The first is Asymmetric Video Encoding Adaptation (*AVA*) that sets the rendering settings to the best values and only adapts video encoding settings (QP). The second is Asymmetric Graphics Rendering Adaptation (*ARA*) that sets the video encoding settings to the best values and only adapts graphics rendering settings (including texture detail and view distance proposed by [4]). The third is called No Adaptation (*NA*) where we fix the rendering and video encoding settings to the best values under the bandwidth limitation (We set *TD*=1 and *QP*=45 to take advantage of the conclusion from UE model). Figure 10(a) shows the bitrates resulting from using all 4 algorithms. The blue curve and red curve which enable changing video settings can adapt to the bandwidth limit very well while the green curve which only enables changing graphics rendering settings is worse. The black dotted line which represents *NA* has to be lower than the minimal bandwidth. Figure 10(b) shows the resulting impairment *I* of the three algorithms. The average *I* for *AVARA* is 11.65 while it is 20.07 for *AVA*, 15.49 for *ARA* and 35.8 for *NA*. Figure 10(c) shows the resulting CMG(3D)-UE score of the three algorithms computed using the user experience model proposed and validated in Section III. The average score for *AVARA* is 4.27 while it is 3.89 for *AVA,* 4.13 for *ARA* and 3.13 for *NA*. We can conclude from Figure 10(b) and (c) that *AVARA* performs much better than *AVA* and *NA* but is slightly better than *ARA*. However, considering the fact that *ARA* includes asymmetric view distance and thereby causes side effects such as dizziness for users if they play 3D games for a long time, *AVARA* is also much better than *ARA*.

VII. CONCLUSION

In this paper, we propose a novel joint optimization approach that makes use of both asymmetric video encoding and asymmetric graphics rendering techniques. It can significantly reduce 3D video encoding bitrate needed for a certain video quality, thereby making it easier to transmit cloud-rendered 3D gaming video over wireless networks to mobile devices. Specifically, we first conduct extensive user studies to develop a user experience model that takes into account both video encoding impairment and graphics rendering impairment. We also develop a model to relate the bitrate of the resulting video with the video encoding settings and graphics rendering settings. By making use of the above two models, we propose an optimization algorithm that can automatically choose the video encoding settings and graphics rendering settings for left view and right view to ensure the best user experience given the network conditions. Experiments conducted using real 4G-LTE network profiles on commercial cloud service demonstrate the improvement in user experience when the proposed optimization algorithm is applied. In the future, we plan to explore the use of more advanced 3D video codecs such as Multiview Video Encoder or Video+Depth 3D Video Encoder in performing joint asymmetric rendering and encoding adaptation to further optimize delivery of Cloud based 3D gaming video.

REFERENCES

[1] http://www.digitaltrends.com/mobile/no-glasses-3d-tablet-released/#!bBUPgr

[2] G. Saygili, C. G. Cihat, and A. M. Tekalp. "Evaluation of asymmetric stereo video coding and rate scaling for adaptive 3D video streaming," *IEEE Transactions on Broadcasting* 57.2 (2011): 593-601.

[3] S. Valizadeh, A. Maryam, and N. Panos. "Bitrate reduction in asymmetric stereoscopic video with low-pass filtered slices." In Proc. of *2012 IEEE International Conference on Consumer Electronics* (ICCE), pp. 170-171.

[4] Y. Lu, Y. Liu, and S. Dey, "Modeling and Optimizing Cloud Mobile 3D Display Gaming User Experience by Asymmetric Graphics Rendering", *IEEE Journal of Selected Topics in Signal Processing, Volume 9, Issue 3, pp. 1-16, Apr. 2015*

[5] B. Vankeirsbilck, T. Verbelen, D. Verslype, N. Staelens, F. De Turck, P. Demeester, and B. Dhoedt. "Quality of experience driven control of interactive media stream parameters." In *Proc. Of IFIP/IEEE International Symposium on Integrated Network Management (IM 2013),* , pp. 1282-1287.

[6] S. Wang, S. Dey, "Adaptive Mobile Cloud Computing to Enable Rich Mobile Multimedia Applications", *IEEE Transactions on Multimedia,* VOL. 15, NO. 4, Jun. 2013.

[7] Z. Ma, M. Xu, Y. F. Ou, and Y. Wang. "Modeling of rate and perceptual quality of compressed video as functions of frame rate and quantization stepsize and its applications." *IEEE Transactions on Circuits and Systems for Video Technology*, 22, no. 5 (2012): 671-682.

[8] http://www.planeshift.it/

[9] ITU-T Rec. G.107, "The E-model, a computational model for use in transmission planning," Mar, 2005.

[10] http://itunes.apple.com/app/speedtest-net-mobile-speed/id300704847?mt=8

[11] http://en.wikipedia.org/wiki/Branch_and_bound

[12] Amazon Web Service, http://aws.amazon.com