Optimizing Cloud Mobile 3D Display Gaming User Experience by Asymmetric Object of Interest Rendering

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

***Abstract*— 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. However, rendering 3D games on mobile devices requires high computational power and battery and thus may restrict users from enjoying true 3D experience for a long time. 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 wirelessly to mobile devices with 3D displays. However, with the significantly higher bitrate requirement for 3D video, ensuring user experience may be a challenge considering the bandwidth constraints and fluctuations of mobile networks. In this paper, we propose a novel asymmetric Object of Interest (OOI) rendering approach, which adapts the rendering richness of different objects according to their importance in order to reduce the video encoding bitrate needed while maintaining a satisfactory video quality, thereby making it easier to transmit the 3D game video over wireless network. Specifically, we first develop a model to quantitatively measure the user experience by different OOI rendering settings. We also develop a model to relate the bitrate of the resulting video with the changes of different OOI Rendering settings. We further propose an optimization algorithm which uses the above two models to automatically decide the optimal OOI 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, subjective testing, asymmetric object of interest rendering**

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 glasses free 3D tablet was released in December 2013; and by August 2014, there are already more than 10 different brands of glasses 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 glasses 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 was introduced earlier [10][11]. 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 which 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. Related works have been done to address this problem which we will discuss later in Section II. However, they all have their limitations. In this paper, we propose a new technique, called *Asymmetric Object of Interest (OOI) Rendering*, which enables the rendering engine to set higher rendering quality for important objects and setting lower rendering quality for unimportant objects, generating video frames that can be encoded with lower bit rates while preserving high user experience. The term *Asymmetric* indicates that the rendering quality of the objects in the left and right views may be set differently. To quantitatively measure the user experience by different Asymmetric *OOI* Rendering settings, we performed subjective tests and developed a user experience model. Moreover, we also develop a rate model to relate the bitrate of the resulting video with the changes of different *OOI* rendering settings. By taking use of the above two models, we further propose an optimization algorithm to automatically decide the Asymmetric *OOI* Rendering settings

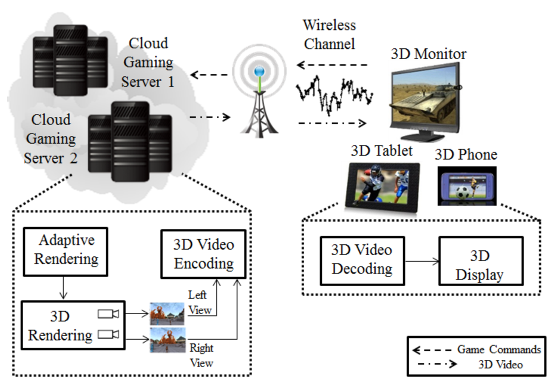

Figure 1. Architecture of Cloud based Mobile 3D Display Gaming System

Figure 2. Object of Interest Example

for left view and right view to ensure the best user experience given current network conditions.

The rest of the paper is organized as following: Section II reviews related works. Section III introduces details about Asymmetric *OOI* Rendering and illustrates its potential to enhance user experience. In Section IV, we propose a user experience model to quantitatively measure the user experience according to different Asymmetric *OOI* Rendering settings. In Section V, we propose another model for the relationship between Asymmetric *OOI* Rendering settings and video bitrate. Section VI proposes an optimization algorithm to automatically choose the best Asymmetric *OOI* Rendering setting given certain mobile network conditions. Section VII shows the experimental results using our CMG(3D) prototype hosted on Amazon Web Service and using commercial cellular network profiles. Section VIII proposes future work and concludes the paper.

II. Related Work

In CMG(3D) framework, the objective is to reduce video bitrate while 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. From previous subjective studies [2][3], we conclude that we can potentially use asymmetric video encoding to reduce 3D video bit rate in the range of 10%-30% without compromising overall video quality. However, the above may not be enough to stream high bit rate high motion 3D rendered gaming video over fluctuating wireless networks. As an alternative approach, we proposed in our previous work asymmetric graphics rendering [4] which enables rendering engine to choose different texture detail or view distance settings for left view and right view to optimize user experience, which proves to achieve a 20%-60% bandwidth saving with acceptable impairment. However, the asymmetric view distance technique might cause the effect that certain objects would appear in one view but not in the other view since we may set different view distances for the two views. This kind of artifact might cause game players to feel fatigue or dizzy when they play the game for a long time. The technique proposed in this paper is a new complimentary approach for the asymmetric graphics rendering technique [4] by allowing different rendering quality for different objects depending on their importance. In this technique, we use texture detail as the only rendering parameter that can be adapted, instead of view distance. Therefore, the problem of an object being rendered in only one view but not in the other will not be present in this new technique.

III. Asymmetric Object of Interest Rendering

Inspired by region-of-interest (*ROI*) based video encoding [5] which allocates more encoding bits to the more important regions of a video frame and fewer bits to the less important ones and therefore reduce the bit rate for the same perceptual quality, we propose similar approach for graphics rendering. While rendering a video frame, we can increase the richness of rendering for the more important objects while decreasing the rendering richness for other less important objects, thereby potentially reducing the video content complexity and hence the video bit rate needed to encode the rendered frame while maintaining the perceived video quality. We call this technique Object of Interest (*OOI*) rendering. The term richness of rendering includes several factors such as texture detail, realistic effect, scene complexity, etc. In this paper we focus on texture detail, and the other factors could be researched later as future work. The texture is defined as an image used to put on top of the game objects when doing rendering. 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. In addition, because our use case is true 3D gaming where we will render two views, and these two views will be seen by the user's two eyes separately, we propose a technique called Asymmetric *OOI* rendering, where the rendering richness of an object for one view can be different from the other. Figure 2 shows an example, where the left view (left image) has all the objects rendered with High texture detail, while for the right view , only the boat controlled by the user (the black boat) and the enemy boats (the yellow boats) are rendered with High texture detail and other objects are rendered using Medium texture detail. Our previous study [4] has shown that by lowering texture detail, video bitrate can be lowered tremendously. However, it also influences user experience, thus we need to develop a model to quantitatively measure the user experience when the proposed *AOOI* rendering is performed.

IV. User Experience Model

We define Cloud Mobile 3D Display Gaming Mean Opinion Score (CMG(3D)-MOS) as a measurement metric for CMG(3D)-UE. To derive the formula for CMG(3D)-MOS with Asymmetric *OOI* rendering, we follow the same framework we developed in our previous work [4], where the user experience is formulated using a set of impairment functions including the impairment functions for rendering, encoding and network. We duplicate the functions here:

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

$$R = 100 - I \quad (2)$$

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 various factors including rendering, encoding and network factors. However, in this paper, we will only study a special situation where the impairment is only caused by rendering and there is no impairment due to encoding and network, under the assumption that 1) we can adapt the rendering setting effectively such that the resulting video bit rate will not exceed the available bandwidth and cause network congestion; 2) we will set sufficiently high encoding quality on the encoder (for example, set a low quantization parameter, QP). The general situation where rendering, encoding and network impairments simultaneously occurring will be studied in the future.

Moreover, although we follow the same framework as [4], the model proposed in this paper is different since we will consider assigning different texture details to different objects while in [4] all the objects are assigned the same texture detail. Since in the approach presented in this paper, we only consider to adapt the texture detail of the left and right views, the formula of R factor (Equation (2)) can be re-stated as:

$$R = 100 - I_R = 100 - f(TD_L, TD_R)$$

in which I_R, the rendering impairment, is a function of the texture detail level of the left view (TD_L) and right view (TD_R).

Furthermore, in the proposed Asymmetric *OOI* rendering approach, we will adapt the rendering richness (more specifically, texture detail) for each individual object depending on its importance; hence, we propose the following equation to model I_R

$$I_R = \sum_{i=1}^{K} w_i p_i I_{TD}\left(TD_{L\_i}, TD_{R\_i}\right) \tag{3}$$

in which K represents the total number of objects in the game frame and I_{TD} function describes different impairment values caused by the different texture detail settings on the left and right views. In Equation (3), i is the object index, p_i means the percentage of pixels that the object i occupies within a game frame and w_i means the object weight which is related to the importance of the object i. Equation (3) can be interpreted as: the overall rendering impairment I_R is a weighted average of impairment caused by each object (I_R (TD_{L_i}, TD_{R_i})), considering the importance of this object (w_i) and how much space this object occupies in the game frame (p_i). The value of p_i can be conveniently extracted from the game engine and the importance of each object w_i is a fixed parameter for each object and can be derived through subjective test. For simplicity but without the loss of generality, in this paper, we will discuss a special scenario, where all the objects can be divided into two groups and each group of objects have the same importance. We define the following two categories of objects: 1) Key Objects (*KO*) and 2) General Objects (*GO*). *KO* is defined as the objects which is important to the players and will attract a lot of attention. For example, in the game Broadsides, the player's boat and enemy boats are *KO*. *GO* is

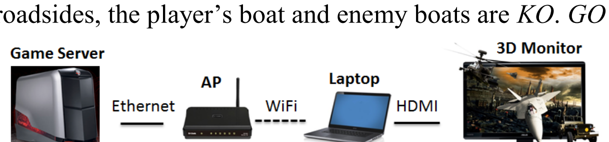

Figure 3. Testbed for Subjective Experiments

defined to be the objects which do not influence the execution of game logic, and eliminating them will not severely affect the playability of the game. For example, trees, rocks, ports, etc. Thus, by replacing w_i with w_{KO} and w_{GO}, Equation (3) can be rewritten as the following.

$$I_R = \sum_{i=1}^{K_1} w_{KO} p_i I_{TD}\left(TD_{L\_i}, TD_{R\_i}\right) + \sum_{j=1}^{K_2} w_{GO} p_j I_{TD}\left(TD_{L\_j}, TD_{R\_j}\right) \tag{4}$$

with the constraint of

$$\sum_{i=1}^{K_1} w_{KO} p_i + \sum_{j=1}^{K_2} w_{GO} p_j = 1 \tag{5}$$

$$\frac{w_{KO}}{w_{GO}} = c \tag{6}$$

In Equation (4), K_1 and K_2 represent the total number of *KO* and *GO* objects respectively, w_{KO} and w_{GO} mean the importance weight for *KO* and *GO* respectively. Moreover, since w_{KO} and w_{GO} are fixed parameters, we can scale them by the same factor without changing the relative importance relation between *KO* and *GO*, such that the constraint Equation (5) is met. In Equation (6), we propose a parameter c to represent the relative importance between *KO* and *GO*. Equation (5) and (6) are used to facilitate our further development. In the following, we performed subjective tests to first decide I_{TD} function and the value of c, so that all the model parameters can be obtained. We also validated the model through subjective test results.

A. Subjective Test settings

Figure 3 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 a first person shooting game Broadsides [7]. To investigate how asymmetric texture detail affect user experience, we set the video encoding parameter QP to 25 which leads to sufficiently high encoding video quality and set network bandwidth to be sufficiently large so that only texture detail can cause impairment. We then invited 16 UCSD students (11 male, 5 female; aged 18-27) 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 graphics rendering factors

TABLE I. GRAPHICS RENDERING FACTOR SETTING

Factors	*Experiment Values*
Texture Detail(Down Sample)	High(0) Medium(2) Low(4)

TABLE II. 3D GRAPHICS QUALITY AND CRITERION FOR I_R

I_R	*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. SUBJECTIVE TEST RESULTS: AVERAGE I_{TD} SCORES FOR DIFFERENT TEXTURE DETAIL COMBINATIONS

Texture Detail Combination	H-H	H-M	H-L	M-M	M-L	L-L
$\boldsymbol{I_{TD}}$	0	18.52	43.35	22.5	24.45	40.5

according to Table I independently for each view. Once a combination of rendering factors is set, we ask the testers to play the game for 1 minute and evaluate the graphics rendering 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 boat, looking for an object, talking to an NPC, etc.).

B. *Impairment Function Derivation and Validation*

In order to derive the I_{TD} function, firstly we change texture details for all the objects together, making all of them to be the same. In this case, because the I_{TD} for each object is the same and because of the constraint Equation (5), Equation (4) can be re-stated as

$$I_R=\left(\sum_{i=1}^{K_1} w_{KO} p_i + \sum_{j=1}^{K_2} w_{GO} p_j\right) I_{TD}\left(TD_L, TD_R\right) = I_{TD}\left(TD_L, TD_R\right) \quad (7)$$

For all the combinations of TD_L and TD_R, we ask the subjects to give evaluation on I_R; and the average evaluation will be used to derive $I_{TD}(TD_L, TD_R)$ function. The results are listed in Table III.

Secondly, in order to derive coefficient c, we change all the KO's texture details together (to be the same) and all the GO's texture detail together (to be the same) to perform another subjective test. In this case, Equation (4) will become:

$$\begin{aligned} I_R = {} & w_{KO} I_{TD}\left(TD_{L\_KO}, TD_{R\_KO}\right)\sum_{i=1}^{K_1} p_i \\ & + w_{GO} I_{TD}\left(TD_{L\_GO}, TD_{R\_GO}\right)\sum_{j=1}^{K_2} p_j \end{aligned} \quad (8)$$

By plugging in constraint Equations (5) and (6), we can obtain the expression for coefficient c:

$$c=\frac{\sum_{j=1}^{K_2} p_j I_{TD}\left(TD_{L\_GO}, TD_{R\_GO}\right) - I_R \sum_{j=1}^{K_2} p_j}{\left(I_R \sum_{i=1}^{K_1} p_i - \sum_{i=1}^{K_1} p_i I_{TD}\left(TD_{L\_KO}, TD_{R\_KO}\right)\right)} \quad (9)$$

According to Equation (9), for any combination of texture details for KO and GO, we can use $I_{TD}(TD_{L\_GO}, TD_{R\_GO})$ and $I_{TD}(TD_{L\_KO}, TD_{R\_KO})$ values computed with Table III, together with the I_R value given by the subjects, to derive the coefficient c. Figure 4(a) shows the results where x-axis and y-axis represent the denominator and numerator of Equation (9) respectively. We can use linear regression method to determine the value of c. For the game Broadsides, c=3.78.

Note that in the above subjective derivation, p_i for each object i for each video frame is different and the user plays the game for 1 minute which includes 25*60=1500 frames. So we use the average p_i value in the derivation process.

Thirdly, after getting c, we then change the texture detail settings for each object independently and ask the testers to

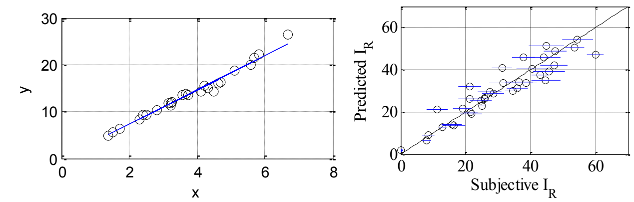

Figure 4. (a) left, Derivation for c
(b) right, Validation of I_R with blue line showing 95% confidence interval

give scores for I_R in order to validate the derived model. Figure 4(b) is the scatter plot of the relationship between subjective and predicted I_R. The correlation is 0.93. Thus, we can conclude that the proposed I_R model (including Equations (1), (2), (4), (5) and (6) and Table III) will lead to high modeling accuracy.

V. BITRATE MODEL

In this section, we develop a bitrate model which estimates the encoding bitrate of a view from the texture detail settings of each object in this view.

Several techniques have been proposed to model the bitrate of encoded video as a function of the video encoding parameter quantization step q and the video frame rate t. For example, in [8], Ma et al. proposed Equation (10).

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

in which q_{min} and t_{max} parameters are chosen based on the application and R_{max} is the actual bitrate when encoding a video at q_{min} and t_{max}. Coefficients a and b are model parameters which depend on the content of the video. The authors in [8] further proposed a method to estimate a and b based on content features shown in Equation (11) and (12).

$$\left[a \;\; b \;\; R_{\max}\right]^T = B\left[1 \;\; \mu_{FD} \;\; \mu_{MVM} \;\; \frac{\mu_{MVM}}{\sigma_{MDA}}\right]^T \quad (11)$$

$$B = \begin{bmatrix} 1.1406 & -0.0330 & -0.0611 & 0.1408 \\ 0.4462 & 0.0112 & 0.0680 & -0.0667 \\ 0.1416 & -0.0008 & -0.0001 & -0.0036 \end{bmatrix} \quad (12)$$

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.

We extend the model proposed in [8] so that we can estimate the resulting video encoding bit rate from the texture details used during rendering of the video. To achieve the above, we develop methods to predict μ_{FD}, μ_{MVM} and σ_{MDA} from the texture detail settings; and further predict the bit rate using Equations (10), (11) and (12). Since in this work we assume there will be no video impairment due to encoding, we fix the values for the quantization step size q and the frame rate t. In the following, we performed several experiments to derive the relationship between texture detail of each object and content features (μ_{FD}, μ_{MVM} and σ_{MDA}). Because the number of objects is not fixed and each object occupies different number of pixels in the view, therefore it is difficult to derive the relationship directly. Hence, instead of using the texture detail of every single object as variable, we use the percentage of pixels with certain texture detail as the variable to predict content features (μ_{FD}, μ_{MVM} and σ_{MDA}). For example, for game Broadsides, it has High, Medium and Low, three texture detail settings in total, hence we propose to model a relationship between the percentage of pixels with each texture detail setting and μ_{FD}. Further, because the sum of the three percentages equal 100, we can use only the percentage of High texture detail pixels and Low texture detail pixels as variables.

In addition, as the content features can always be calculated after the video frame is encoded, next we will study how these content features will vary according to different texture detail settings. Firstly we define a reference case, where we set all objects' texture detail to be Medium. Then we performed the following experiment in which we captured 40 video clips of players' playing the game Broadsides in the same game scene, performing the same tasks with the same time period, but with different texture details of each object. We firstly set all texture detail to be Medium (the reference case) and calculate μ_{FD}, μ_{MVM} and σ_{MDA} associated with this reference case. Secondly, we change texture details of some objects from Medium to be High so that the entire game scene only consists High and Medium texture details and calculated μ_{FD}, μ_{MVM} and σ_{MDA} again. We studied the relationship between the percentage of High texture detail pixels and content features. Thirdly, we did the same experiment but instead of changing to High texture detail, we change to Low texture details. Figure 5 shows the results.

X-axis of Figure 5(a)(c)(e) show percentage of High texture detail pixels and x-axis of Figure 5(b)(d)(f) show percentage of Low texture detail pixels. Y-axis of all figures shows the ratio of the corresponding content feature over content feature of the reference case. We can conclude from the above plots that μ_{MVM} and σ_{MDA} are almost independent on texture details of each object (shown in Figure 5(c)~(f)).

$$\mu_{\text{MVM}}\left(\sum_{TD_{L\_i}=H} p_i, \sum_{TD_{L\_k}=L} p_k\right) = \mu_{\text{MVM}}\left(0,0\right) \tag{13}$$

$$\sigma_{MDA}\left(\sum_{TD_{L\_i}=H} p_i, \sum_{TD_{L\_k}=L} p_k\right) = \sigma_{MDA}\left(0,0\right) \tag{14}$$

However, μ_{FD} has an almost linear relationship with the percentage of High texture detail pixels or Low texture detail pixels (shown in Figure 5(a)(b)). Thus we propose Equation (15) to relate μ_{FD} with the texture detail settings.

$$\frac{\mu_{FD}\left(\sum_{TD_{L\_i}=H} p_i, \sum_{TD_{L\_k}=L} p_k\right)}{\mu_{FD}\left(0,0\right)} = H\left[1, \sum_{TD_{L\_i}=H} p_i, \sum_{TD_{L\_k}=L} p_k\right]^T \tag{15}$$

For game Broadsides, according to Figure 5(a) and 5(b), we can use linear regression method to derive H= [1 -0.21 0.37]. We validate Equation (13) through the rest of the 40 video clips which contains High, Medium, Low texture detail settings in the same view. The validation results are shown in Figure 6. The correlation between the predicted content feature μ_{FD} and the actual μ_{FD} is 0.95.

Hence, we claim that the content feature μ_{FD} can be accurately predicted using Equation (13). Moreover, from

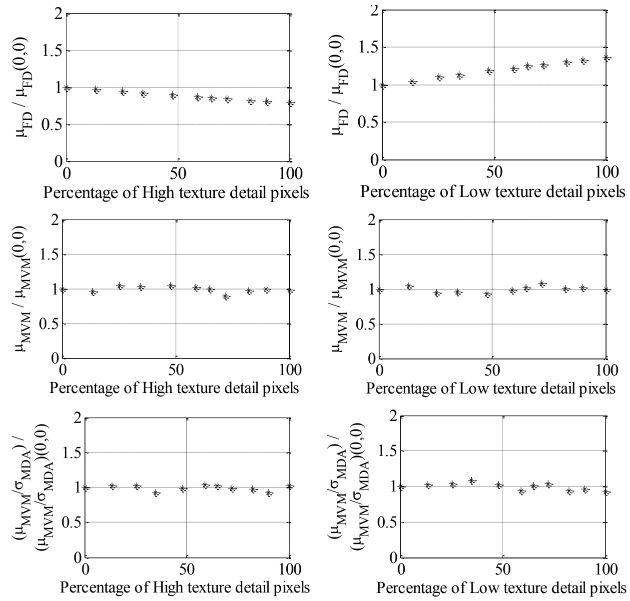

Figure 5. (a) top left, relationship between percentage of High texture detail pixels and μ_{FD} (b) top right, relationship between percentage of Low texture detail pixels amd μ_{FD} (c) middle left, relationship between percentage of High texture detail pixels and μ_{MVM} (d) middle right, relationship between percentage of Low texture detail pixels and μ_{MVM} (e) bottom left, relationship between percentage of High texture detail pixels and μ_{MVM}/σ_{MDA} (f) bottom right relationship between percentage of Low texture detail pixels and μ_{MVM}/σ_{MDA}

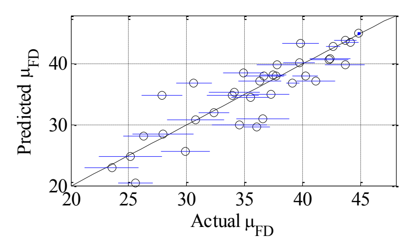

Figure 6. Validation of μ_{FD} with blue line showing 95% confidence interval

Given:
1) Network bandwidth BW
2) Texture Detail bound TD_{min} and TD_{max}
3) Percentage of pixels for each object p_i
4) Importance weight ratio c
5) Current content features μ_{FD}, μ_{MVM} and σ_{MDA}

Find:
The optimal graphics rendering factors, TD_{L_i} and TD_{R_i}, for each object i to minimize I_R

$$I^{opt} = \min\left(I_R\right) = \min\left(\begin{array}{l}\sum_{i=1}^{K_1} w_{KO} p_i I_{TD}\left(TD_{L\_i}, TD_{R\_i}\right) + \\ \sum_{j=1}^{K_2} w_{GO} p_j I_{TD}\left(TD_{L\_j}, TD_{R\_j}\right)\end{array}\right) \tag{16}$$

s.t.

$$\sum_{i=1}^{K_1} w_{KO} p_i + \sum_{j=1}^{K_2} w_{GO} p_j = 1, \quad \frac{w_{KO}}{w_{GO}} = c \tag{17}$$

$$TD_{\min} \le TD_{L\_i} \le TD_{R\_i} \le TD_{\max} \tag{18}$$

$$R_L\left(\sum_{TD_{L\_i}=H} p_i, \sum_{TD_{L\_k}=L} p_k\right) + R_R\left(\sum_{TD_{R\_i}=H} p_i, \sum_{TD_{R\_k}=L} p_k\right) \le BW \tag{19}$$

Figure 7. Problem Formulation

figure 5 we can see the value of content feature μ_{MVM} and σ_{MDA} are not dependent on the texture detail settings. These three content feature Equations (13), (14) and (15) together with Equations (10), (11) and (12) complete our bit rate model.

VI. OPTIMIZATION ALGORITHM

In Section IV and V we have proposed 1) a user experience model which models cloud mobile 3D display gaming user experience as a function of texture detail settings of each object and 2) a bitrate model which estimates video bitrate needed to encode the rendered video as a function of the texture detail settings of each object used during rendering. In this section, we combine these two models so that by selecting proper texture detail settings of each object we can find an optimal solution for maximizing user experience given a network bandwidth limit. Figure 7 shows the problem formulation. Because according to Equation (1) and (2), maximizing CMG(3D)-UE is equal to minimizing I, and in this research, I only includes I_R, we set optimization objective to be minimizing I_R. TD_{min} and TD_{max} are the minimum and maximum boundaries of texture details being used for a game. So if the rendering engine uses texture detail values of High, Medium, Low, and we assign 0 to be High, 1 to be Medium and 2 to be Low, then $TD_{min} = 0$, $TD_{max} = 2$. The percentage of pixels object i occupies, p_i, can be obtained through game information and coefficient c has already been studied in Section IV.

In order to solve the above problem, we propose an Asymmetric Object of Interest Rendering Adaptation (*AOOIA*) 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; 2) percentage of pixels for each object p_i and current content features (μ_{FD}, μ_{MVM} and σ_{MDA}). We use the p_i value and the content features in the current time period with Equation (13) to estimate the bitrate consumption in the next time period. The output of the algorithm will be the optimal texture setting for each object.

This problem is actually a variant of classic *knapsack* problem [9], in which the bit rate consumed by each object can be interpreted as the *weight* of this object; and the rendering impairment caused by each object can be regarded as the *price* of each object. The problem can then be restated as, for total K objects, select the optimal texture detail setting for each object, such that the total *weight* does not exceed the limit (BW), and the total price is minimized. The difference is that for *knapsack* problem, each object has a fixed *weight* and fixed *price*, while in our problem different texture details result in different *weight* and the resulting *price* even changes depending on texture detail settings of not only itself but also other objects. Thus, the widely used classical dynamic programming algorithm cannot be applied in this case. Further, considering the fact that the algorithm needs to be executed in as little time as possible as CMG(3D) application has a real time execution requirement, we propose *AOOIA* which is an approximation algorithm based on greedy approach [9], and is shown in Figure 8.

The underlying principle of the algorithm (Figure 8) is as follows: initially we set the texture detail of all the objects to be Low in both left and right views. Then we keep adjusting the texture detail of objects using a while loop, as long as the total bit rate does not exceed the bit rate budget (BW). During each iteration, the algorithm will firstly iterate over all the objects and for each object i, the algorithm computes the possible degradation in its rendering impairment ($\triangle\_I_i$) and the possible increase in the consumed bandwidth ($\triangle\_BW_i$), if we set its texture detail in one view to be one level higher. Among all the objects, the algorithm will choose the one with the highest ratio of $\triangle\_I_i / \triangle\_BW_i$. The algorithm will stop when 1) it reaches the maximum iteration rounds or 2) there is no more bandwidth available or 3) all objects set texture detail to be High. The proposed *AOOIA* algorithm is runs within 10ms in a computer with a dual-core i7 processor and thus can meet this real time execution requirement.

Algorithm Asymmetric Object of Interest Rendering Adaptation (*AOOIA*)

Input:
1) Network bandwidth BW
2) Texture Detail bound TD_{min} and TD_{max}
3) Percentage of pixels for each object p_i
4) Content features μ_{FD}, μ_{MVM} and σ_{MDA}
5) Importance weight ratio c

Output: I^{OPT}, $TD_{L\_i}{}^{OPT}$, $TD_{R\_i}{}^{OPT}$

Step 1: For each object i, set both $TD_{L\_i}{}^{OPT}$ and $TD_{R\_i}{}^{OPT}$ to be TD_{max}, calculate the current bandwidth needed BW_{cur} and I_{cur}
$BW_{pre} = BW_{cur}$
$I_{pre} = I_{cur}$
Step 2: While ($BW - BW_{cur} > 0$) && (round < round_limit) && !(all TD == 0)
  For i=1; i<K; i++
    $\max(TD_{L\_i}{}^{OPT}, TD_{R\_i}{}^{OPT}) = \max(TD_{L\_i}{}^{OPT}, TD_{R\_i}{}^{OPT}) - 1$
    calculate BW_{cur} and I_{cur}
    $\triangle\_BW_i = BW_{cur} - BW_{pre}$
    $\triangle\_I_i = I_{pre} - I_{cur}$
    $\max(TD_{L\_i}{}^{OPT}, TD_{R\_i}{}^{OPT}) = \max(TD_{L\_i}{}^{OPT}, TD_{R\_i}{}^{OPT}) + 1$
  Endfor
  Find i which has the maximum value of $\triangle\_I_i / \triangle\_BW_i$
Step 3: Set the change of TD setting of object i
  $\max(TD_{L\_i}{}^{OPT}, TD_{R\_i}{}^{OPT}) = \max(TD_{L\_i}{}^{OPT}, TD_{R\_i}{}^{OPT}) - 1$
 Endwhile
Step 4: **Return** I^{OPT}, $TD_{L\_i}{}^{OPT}$, $TD_{R\_i}{}^{OPT}$

Figure 8. *AOOIA* Algorithm Diagram

VII. EXPERIMENTAL RESULTS

In this section, we report on experiments conducted using a commercial cloud service to verify the performance improvement by applying the proposed Asymmetric *OOI* Rendering Adaptation (*AOOIA*) technique. We use the same testbed as shown in Figure 3, except we implement our CMG(3D) system, including the *AOOIA* algorithm, on Amazon Web Service (AWS) servers. 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 [12] to record the bandwidth around UCSD campus. Figure 10(a) shows a sample LTE trace, which is emulated using the network emulator in our testbed. In addition, for comparison reasons, we also implemented two other algorithms called Asymmetric Adaptation (*AA*) which is based on our previous work [4] and Symmetric Adaptation (*SA*). *AOOIA*, *AA* and *SA* all follow the same algorithm diagram shown in Figure 8 except that *AA* does not enable Object of Interest, which means it can only set the texture detail to be the same for all the objects in the same view and *SA* further disables asymmetric texture detail setting so that all the objects in two views have the same texture detail. Figure 10(a) shows the bitrates resulting from using all 3 algorithms. The blue dash-dot curve which represents the bitrate of *SA* is closer to the black dotted curve (the bandwidth trace) than the green dashed curve which stands for the bit rate of *AA;* but the red solid curve which represents *AOOIA* is the closest among the three. The above indicates that *AAOIA* can adapt to the network bandwidth fluctuations the best, with its finer granularity capability of adapting the texture detail of each object in each view, followed by *AA* which can only adapt each view independently but not each object, followed by *SA* which can adapt the texture detail but for all the objects of both views.

Figure 10(b) shows the resulting rendering impairment I_R of the three algorithms. The average I_R for *AOOIA* is 13.62 while it is 21.89 for *AA* and 26.10 for *SA*. 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 IV. The average score for *AOOIA* is 4.12 while it is 3.83 for *AA* and 3.63 for *SA*. Thus, the results in Figure 10(b) and (c) prove that *AOOIA* can lead to much less impairment and thus much better user experience as measured by CMG(3D) compared with the two previously proposed algorithms [4].

VIII. Conclusion

In this paper, we propose a novel Asymmetric (*OOI*) Rendering approach which 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 develop a model to quantitatively measure the user experience by different *OOI* rendering settings. We also develop a model to relate the bitrate of the resulting video with the changes of different *OOI* Rendering settings. By making use of the above two models, we further propose an optimization algorithm to automatically decide the optimal *OOI* rendering settings for objects in left and right views to ensure the best user experience given the network conditions. Experiments conducted using real 4G-LTE network profiles and commercial cloud service demonstrate the improvement in user experience when the proposed optimization algorithm is applied. In the future, we are interested in including more impairment categories, i.e. network delay impairment and video encoding impairment to make the model more general and robust. We also plan to implement the proposed system using some more advanced 3D video codecs such as Multiview Video Encoder or Video+Depth 3D Video Encoder.

IX. Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CCF-1160832.

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'12), pp. 170-171.

[4] Y. Lu, Y. Liu, and S. Dey. "Enhancing Cloud Mobile 3D display gaming user experience by asymmetric graphics rendering." In Proc. of *2014 IEEE International Conference on Computing, Networking and Communications* (ICNC'14), pp. 368-374.

[5] Y. Liu, Z. G. Li, and Y. C. Soh. "Region-of-interest based resource allocation for conversational video communication of H. 264/AVC." *IEEE Transactions on Circuits and Systems for Video Technology,* 18.1 (2008): 134-139.

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

[7] Broadsides, http://cse125.ucsd.edu/cse125/2012/cse125g1/

[8] 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. 5 (2012): 671-682.

[9] S. Martello, and T. Paolo "Knapsack problems." New York: Wiley, 1990.

[10] Y. Liu, S. Wang and S. Dey, "Modeling, Characterizing, and Enhancing User Experience in Cloud Mobile Rendering." In Proc. of *2012 IEEE International Conference on Computing, Networking and Communications* (ICNC'12), pp.739-745

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

[12] http://itunes.apple.com/app/speedtest-net-mobile-speed/id300704847

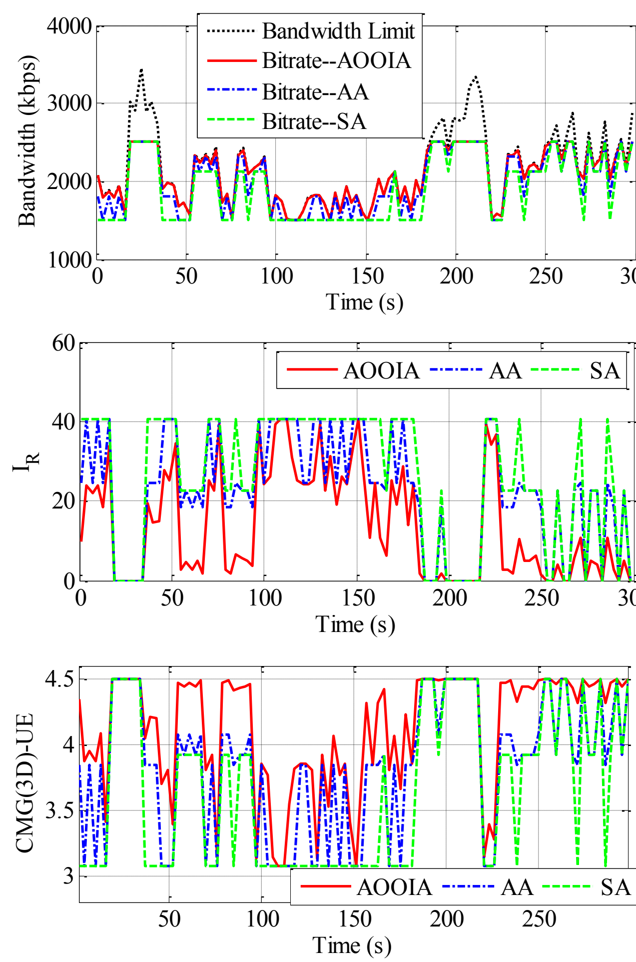

Figure 10. (a) top, network bandwidth limit and bandwidth consumption of three algorithms (b) middle, I_R of three algorithms (c) bottom, CMG(3D)-UE of three algorithms