Efficient Bitrate Reduction Using A Game Attention Model in Cloud Gaming

Hamed Ahmadi[1], Sepideh Khoshnood[1], Mahmoud Reza Hashemi[1], Shervin Shirmohammadi[1,2]

[1] Multimedia Processing Laboratory (MPL), School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Iran, [ha.ahmadi | s.khoshnood | rhashemi | sshirmohammadi]@ut.ac.ir

[2] Distributed and Collaborative Virtual Environments Research (DISCOVER) Lab, School of Eelctrical Engineering and Computer Science, University of Ottawa, Canada, shervin@discover.uottawa.ca

***Abstract*— Cloud gaming is a new promising service which combines the advantages of cloud computing and online gaming. The concept of cloud gaming is to render a game on a cloud server and stream its scenes to the player as a video sequence over a broadband connection. In order to attract more customers and increase profitability, one needs an efficient and scalable framework to address the heterogeneity and constraints of client devices as well as the network infrastructure. In this paper, we introduce the concept of Game Attention Model (GAM), which is basically a hybrid visual attention model, as a means for reducing the bit rate of the streaming video more efficiently. GAM estimates the importance of each macro-block in a game frame from the player's perspective and allows encoding less important macro-blocks with lower bit-rate. Subjective assessment shows that by integrating this model into a cloud gaming framework, it is possible to decrease the required bit rate by nearly 50 percent in average, while maintaining a relatively high user quality of experience.**

Keywords*— *Cloud Gaming; Visual Attention Model; Bit Rate Reduction; Content-aware video coding;

I. INTRODUCTION

With the introduction of fast and reliable core networks and wide-spread availability of broadband internet access, a trend towards moving more and more services away from the end devices to remote data centers has established itself. Cloud gaming is among these services which has rapidly expanded its market among gamers and drawn a lot of attention from researchers [1-3] and businesses. Most recently, Nvidia announced that it has created unique features in its upcoming graphics chips that could make cloud-based gaming much more practical [4].

The concept of cloud gaming is to render a video game on a cloud server and stream the game scenes as a video to game players over a broadband network A cloud gaming framework has been illustrated in Figure 1. In this framework, the user input signals (mouse, keyboard, or game controller events) are sent back to a cloud server to interact with the game application. Cloud gaming has many advantages for users as well as game developers. On one hand, users no longer need to purchase high end graphical hardware to run new games and can play on virtually any device that can decode and display video. On the other hand, developers no longer have to fear software piracy, as the software never leaves the cloud. Furthermore, this approach can reduce development costs by focusing on one specific platform.

Cloud gaming, however, has some limitations. First, it requires a high bandwidth network to simultaneously stream the game as a video sequence to multiple players. For example, OnLive [5] requires a wired network connection with no less than 5Mbps constant bandwidth per player to provide interactive gaming services with a resolution of 720p at 30fps. Second, it is sensitive to network latencies since a long latency seriously impairs the interactive experience of a video game [2]. These restrictions make cloud gaming unavailable to most mobile users and those who have low quality and low bandwidth network access. The goal of this paper is to offer a new model which helps to efficiently decrease the bit rate of the streaming video such that users with limited computational and communication resources can still benefit from this service with acceptable quality.

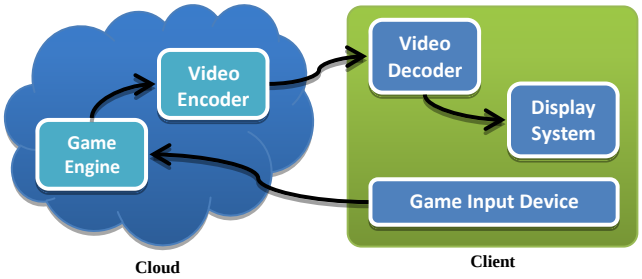

Fig. 1. Cloud gaming concept diagram.

In terms of complexity, this model proposes some moderate load to the cloud side because of its computationally light components and independence of its main blocks that can be run in parallel. Besides, unlike a player's bandwidth the cloud side is scalable, so incurring some additional complexity at the cloud side is justified by the benefit of being able to support more players who have lower bandwidths. It is well-known that cloud gaming today requires very demanding bandwidths from players [6], so any technique that lowers that requirement is of great interest to the gaming industry.

The idea is, for a given game frame, to encode each macro-block of the video based on the "importance" of that macro-block from the player's perspective; i.e., if at a given moment a macro-block is less important for the player, it should be encoded with a lower quality compared to more-important

macro-blocks, leading to a reduced video bitrate without significantly affecting the player's quality of experience. To achieve this, the first step is to find a model to evaluate the importance of different regions of the game frame for a player. Then, we need to determine the corresponding encoding configuration according to the results of the previous step. In this paper, we introduce a conceptual model which estimates the importance of regions of a game frame based on the amount of attention the player would pay to them.

A typical player directs attention to objects within a video frame using both bottom-up, image-based saliency cues and top-down, task-dependent cues [7]. For example, consider an injured player looking for a first aid kit, which is typically a small white box with a red cross logo on it. Any region with high changes in brightness draws the player's bottom-up attention regardless of the task at hand. But since he is looking for a first aid kit, he tries to direct his top-down attention to white areas and/or find a red cross.

Visual attention models try to computationally simulate these neurobiological mechanisms. They have already been used in many applications such as graphics, arts and human computer interaction. For example, it can be used to direct foveated image and video compression and levels of detail in non-photorealistic rendering. In this paper, we use Judd's Support Vector Machine (SVM) based model which has been trained using a large database of eye tracking data [8].

Once the level of importance of a macro-block is determined, it can be used for selecting the best encoder configuration that can lead to the most efficient bit allocation for it. For simplicity and without lack of generality, in this paper we consider the quantization parameter (QP) as the only encoder parameter that can be controlled. To do so, we try to select an appropriate QP value for each region of the video frame. Clearly, a greater quantization step for a macro-block might decrease its quality. Considering the predictive nature of most video encoders this can spread to its spatial and temporal neighbors. However, the fact that the user does not pay much attention to unimportant regions, and the possibility of choosing a better quantization step for important regions, maintains the user's overall quality of experience.

The remainder of this paper is organized as follows. The next section overviews related works. The proposed game attention model is explained in Section 3. Section 4 describes our implementation followed by Section 5 that presents our evaluation results and analysis. Finally, the paper ends with discussion and future works, and concluding remarks in separate sections.

II. RELATED WORKS

As cloud gaming has become more widespread, researchers have shown interest in its different aspects. Jarschel et al. [1] have studied the user-perceived QoE in cloud gaming and have shown that the perceived game experience is not only dependent on the QoS parameters of delay and loss, but also has to be put into context with the content. Chen et al. [2] have analyzed the response latency of two cloud gaming platforms and have proposed a methodology to measure different types of delay in cloud gaming platforms, including network delay, processing delay and play-out delay.

In cloud gaming, a game is rendered on a cloud server and its scenes are streamed to the player as a video sequence. Another kind of game streaming is geometry transmission in which game objects are first streamed to the player in 3D graphics format, and then rendered on the end device. Despite their basic difference, it is possible to utilize some ideas from geometry streaming into cloud gaming. As an instance, in this paper, we use a graphics adaptation scheme for virtual environments using optimized object selection [9]. This scheme uses an object selection and optimization method so that only the most important objects from the perspective of the player's activity are included in the scene and irrelevant or less important objects are omitted.

Cloud gaming is fundamentally a video streaming service on which there has been a great deal of research. For example, [10] has proposed an attention-based spatial adaptation scheme which not only improves the perceptual quality but also saves the bandwidth and computation. They have used a visual attention model to predict, at a given video frame, how much attention viewers pay to each part of the frame.

Visual attention models have been used in diverse applications in which it is essential to understand where humans look in a scene. They measure the conspicuity of a location, or the likelihood of a location to attract the attention of human observers.

In this paper, we offer a novel model to simultaneously utilize the advantages of visual attention model and optimized object selection scheme. Specifically, in each frame of the gameplay, we consider the context of the game and visual saliency features to decide which regions of the frame are more important for the accomplishment of the player's current activity. To the best of our knowledge, no other research has combined visual attention with object priority to reduce cloud gaming bitrate. In fact, how to do ROI analysis in games is not yet fully understood and this paper is an attempt toward this goal.

III. PROPOSED MODEL

In this paper, we introduce a model which helps the encoder to decrease the bit rate of game frames without noticeable loss in quality. This model, referred to hereafter as Game Attention Model (GAM), is responsible for determining the importance of different regions of the current game frame from the player's perspective, so that the encoder would be able to allocate more bit rate to those regions in proportion to their importance, and less bit rate to the less-important regions, resulting in a lower overall bit rate without significant reduction in the quality of experience from the player's perspective.

In order to design a GAM, one should find the factors that affect the importance level of a given region in a game frame. Undoubtedly, the most significant factor is the game logic. Any model that does not consider the game's genre, objectives and players would hardly prove relevant. For example, in a First Person Shooter game, those enemy units which are engaged in a fight with the main character are more important than other

units seen in any given game frame. Another significant factor is the way that the player pays attention to the game frame. Attention is at the nexus between cognition and perception. As mentioned before, the control of the focus of attention may be goal-driven and stimulus-driven which corresponds to top-down and bottom-up processes in human perception, respectively. For example, parts of the frame with sharp intensity changes involuntarily draw the player's attention. Attention models which combine these two forms outperform the ones which have been developed based on only bottom-up or top-down attentions [8].

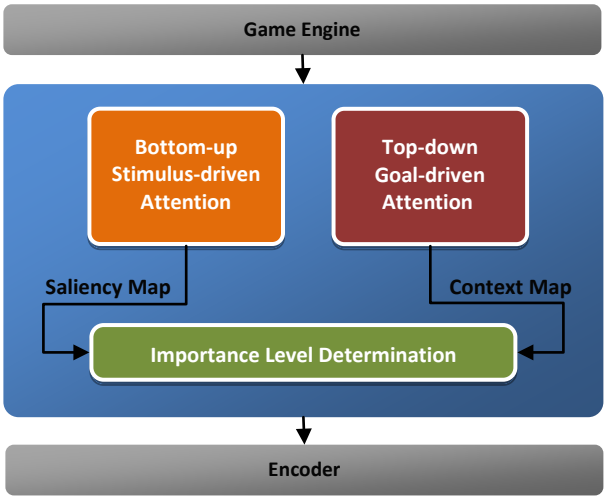

Fig. 2. Proposed Game Attention Model.

In a bottom-up computational model of human attention, typically, multiple low-level visual features such as intensity, color, orientation, texture and motion are extracted from the image at multiple scales. Subsequently, a saliency map is computed for each of the features, and they are normalized and combined in a linear or non-linear fashion into a master saliency map that represents the saliency of each pixel.

The second form of attention, top-down attention, is a more deliberate and powerful one that has variable selection criteria, depending on the task at hand. As an example, if the player's car is running out of fuel, he would pay more attention to the signs to find a gas station.

As illustrated in Figure 2, we have considered both types of attentions in GAM. The bottom-up attention block produces a saliency map. This map, together with a priority map, which comes from the top-down attention block, is fed into an Importance Level Determination block (ILD). ILD is responsible for combining the results of attention blocks in a coherent manner. It can be implemented as easy as a pixel-wise maximum operation or as complex as an intelligent machine learning model.

In summary, GAM consists of four tasks: 1) When the game engine starts rendering the current frame, it provides GAM with a list of current objects and player's activity. This data is fed into the top-down attention model to create the priority map. This extra rendering does not include any texture, shader or light processing, so it can be done even before the main frame has been rendered completely. 2) When the game engine finishes rendering the current frame, it feeds it into GAM, generating a bottom-up attention map. 3) The final attention map for each macro-block is generated by merging the two maps immediately after task 2. 4) Macro-blocks with the same importance are grouped as slices and fed to the encoder using FMO. The QP value of each slice is set based on the importance level of its macro-blocks.

The goal of this paper is to show how helpful GAM would be to reduce the video bit rate in cloud gaming. Hence, we simply selected two recent implementations for each of the two attention blocks in GAM (see Figure 2). Although this might not be the optimal selection, it adequately serves the purposes this paper. This sample implementation is explained in more detail in the next section.

IV. IMPLEMENTATION

In this section, we describe the implementation steps of one example of GAM and its use with an H.264/AVC encoder. We first need to choose an instance for each block in Figure 2. For the bottom-up attention block, we selected Judd's SVM-based model which has been trained using a large database of eye tracking data [8], and for the top-down attention block, we opted the game priority model proposed in [9]. The regions highlighted by these two models are not necessarily overlapped. Therefore, relying on only one of them may result in missing some important regions of the game frame which consequently decreases the user's experience or unnecessarily increases bit rate.

Figure 3 shows how GAM calculates a multi-level importance map from its input game frame. First, the bottom-up saliency map is calculated. This gives a grayscale map which is then converted to a black and white map via a threshold. If the threshold is set too low, some important regions will be missed. If it is set too high, some unimportant regions will be included in the map. Both cases affect the performance of the model. In our work, this threshold is determined experimentally so that the map contains proportionate amount of salient regions. Second, objects in the game frame are prioritized as described in [9]. Subsequently, each object is assigned a gray level according to its priority rank, producing a priority map. In order to keep computational costs down, we use bounding-boxes of game objects. In this paper, we have only considered three priority levels for this map. Finally, the saliency and priority maps are combined via a pixel-wise maximum operation. Hence, we now have an attention map in which each pixel has a value, indicating its importance. Since the smallest unit on which the encoder operates is a macro-block, we divide the attention map into 16x16 blocks and assign a value to each of them. To do so, for each 16x16 block, we assign the maximum importance value of all the pixels in that block as its final level of importance. Note that we do not use averaging operation for this step, because otherwise a macro-block with few important pixels might be treated as an unimportant one due to the averaging effect. To further generalize the model, we consider two weighting factors, so one can control the influence of either saliency or priority map in the final attention map. In our work, we realized experimentally that setting the saliency weight to

half of the priority weight would help the model better partition the macro-blocks according to their importance.

Once the macro-block level priority map is generated, we can set the quantization parameter of each macro-block according to its priority level. In our experiment, we use a three-level map and hence three QP values. The higher the priority level of a macro-block is, the smaller its QP value must be. Figure 3 shows the output map of each block and the final attention map for two sample game frames.

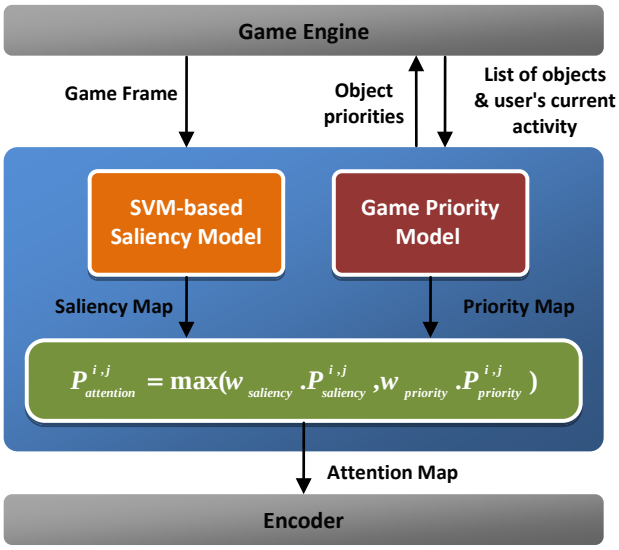

Fig. 3. Our implementation of GAM.

In practice, only the first frames of each GOP might be analyzed to find the appropriate QP values. Also, the aforementioned thresholds are the same for all these frames.

JM V18.4 software [11] is an implementation of H.264/AVC based on which we conduct our test. In order to put macro-blocks of the same importance into separate slices, we activate the Flexible Macro-block Ordering (FMO) tool which is one of several error resilience tools defined in the Baseline profile of this standard.

Similar to Onlive and other cloud gaming systems that offer HD quality games, we choose 720p game frames for our experiment. The suitable level of the Baseline profile at this resolution is 3.1.

V. EVALUATION

Decreasing the QP values of some macro-blocks in the game frame evidently diminishes PSNR and other similar objective quality metrics. But as mentioned before, we expect the user's perceived quality not be affected significantly by degradations in regions in the game frame which are less important to his current gaming context and viewing experience. In order to verify this assumption, we conducted a subjective assessment. First, we gathered 18 game frames from recent popular video games. Then we encoded each frame with and without the help of the proposed GAM model. Finally, we compared the bit rate and objective and subjective quality of the decoded version of those frames.

In this paper, 13 out of 18 game frames were chosen from First Person Shooter games (Call of Duty® and Far Cry®) since fast games are more tolerant towards loss than other game genres [1]. The other five frames were selected from Need For Speed®, Moto Racer and Chicken Invaders, respectively. The former two belong to the racing genre and the latter one to shoot 'em up.

The reference game frames in our experiment were encoded without GAM and hence using a single QP value. In order for the comparison to be fair, we chose the maximum QP value for which one could not detect any distortion in these frames. This maximum QP value was 30 and the average PSNR of the references for this QP value was 36 dB.

Those same frames were once again encoded with the help of GAM. In this case, each frame was encoded with three QP values corresponding to the priority regions that were determined by the two attention components of GAM, namely saliency map and priority map in our implementation. The three QP values were selected to be 30, 35 and 40 for the high, medium and low priority macro-blocks, respectively. To further evaluate the impact of each of the two attention models we encoded each frame once with priority map only with the same three QPs, and another time with saliency map only with a subset of the above QPs. It should be noted that the saliency only results were not assessed in the subjective experiments since their distortion were clearly noticeable, and also because there was a restriction on the duration of each test run according to the standard.

TABLE I. DEMOGRAPHICS OF VIEWERS

Gender				
Male		Female		
67%		33%		
Gaming Experience				
Bad	Poor	Good	Fair	Excellent
33%	13%	27%	14%	13%
Monthly Game Play				
<= 5	6 - 10	11 - 20	21 - 30	> 30
60%	13%	7%	7%	13%
Gaming Platform (Already Played on)				
PC	Console	Tablet	Cellphone	
87%	53%	40%	73%	
Genres (Already Played)				
FPS	TPS	Shoot'em up	Fighting	
50%	36%	33%	56%	
Adventure	Strategy	Sports	Race	
37%	43%	80%	90%	

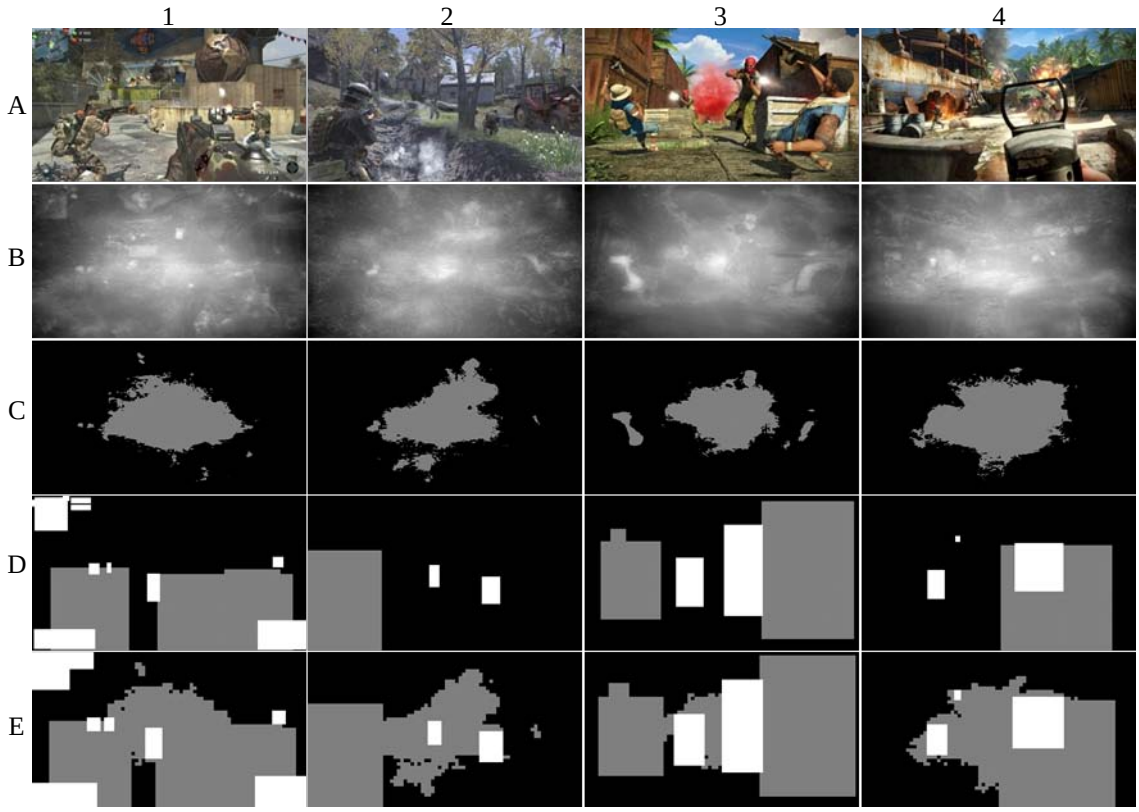

Fig. 4. Results of each block in Figure 3 for four sample game frames. A) Game frames, B) Saliency map, C) Saliency map after applying the threshold, D) Priority map and E) GAM attention map.

The subjective quality of the decoded sequences was assessed using the Double Stimulus Continuous Quality Scale method described in ITU-R Recommendation 500 (ITU-R 1974-1997). Thirty viewers participated in this experiment. All of them were non-experts with no expertise in video processing and quality assessment. Table I shows the demographic profile of the viewers.

The game frames in our study were viewed by each viewer and required twenty minutes of their time in total. The game frames were displayed at their original resolution to prevent any distortions due to scaling operation. The viewing distance was set to four times the screen height as recommended in Rec. ITU-R 812. Viewers were presented with the single-QP and multi-QP decoded sequences randomly with 3 seconds gray display between them. Each pair of sequences was repeated three times. At the end, viewers evaluated the subjective quality of both sequences on a quality scale from 1 to 5 corresponding to "Bad", "Poor", "Fair", "Good" and “Excellent” quality, respectively. They were informed that their evaluations are not necessarily required to be integers and they could choose real numbers. At the beginning of each test session, five “dummy presentations” were introduced. The first one is to familiarize the observers with the setup, and the rest of them to stabilize their opinions.

Table II presents the results of our experiment for the no GAM and the two GAM scenarios. As mentioned before, in one GAM scenario we used only the priority map [9] while in the other one we exploited both saliency and priority maps. This way, we can compare the result of our work (priority + saliency) with previous work (no GAM, and priority only). In Table II, the subjective quality for each scenario is expressed as the mean opinion score (MOS), and the average bit rate (KB), PSNR (dB) and SSIM index for the game frames in our experiment are also shown.

TABLE II. RESULTS OF OUR EXPERIMENT FOR THREE SCENARIOS

	no GAM	**Priority Only**		**Saliency + Priority**	
	1 QP	3 QPs	Change	3 QPs	Change
Bit rate (KB)	79	37.4	-51.6%	39.8	-48.7%
MOS	4	3.67	-8.25%	3.81	-4.75%
PSNR (dB)	36.6	31.5	-14%	31.8	-13%
SSIM	1	0.93	-6.7%	0.97	-2.3%

As we can see from the results, GAM has achieved about 50 percent bit rate reduction on average. Obviously, such a reduction would decrease PSNR and SSIM as it has here. But the very small actual perceived quality reduction of 4.75%, as reported by the subjects in their MOS is a testimony that this distortion has been mostly hidden from the viewers and has not significantly affected their quality of experience with the game.

Between the two GAM scenarios, the subjective quality, PSNR and SSIM are better in "Saliency + Priority" compared to the "Priority only" scenario. Although “Priority Only” achieves 51.6 - 48.7 = 2.9% more bitrate reduction than

"Saliency + Priority", the latter improves the quality as perceived by the players by a factor of 8.25/4.75 = 1.7 (170%), which is significant. Hence, it can be concluded that simultaneously using both of these two maps in GAM will lead to the best balance between reducing bit rate and maintaining quality of experience. Figure 5 shows two versions of one of the game frames used in our evaluations. In this figure, the image on top has been encoded with no GAM using a single QP value of 30. The one at the bottom has been encoded with "Saliency + Priority" using three QP values of 30, 35 and 40 for the high, medium and low priority macro-blocks, respectively. In this image S1, P1, and P2 represent the high quality regions of the saliency map, and the high and medium regions of the priority map, respectively. As expected, frame areas outside these regions have higher compression and hence lower quality, which does not affect the player's gaming experience.

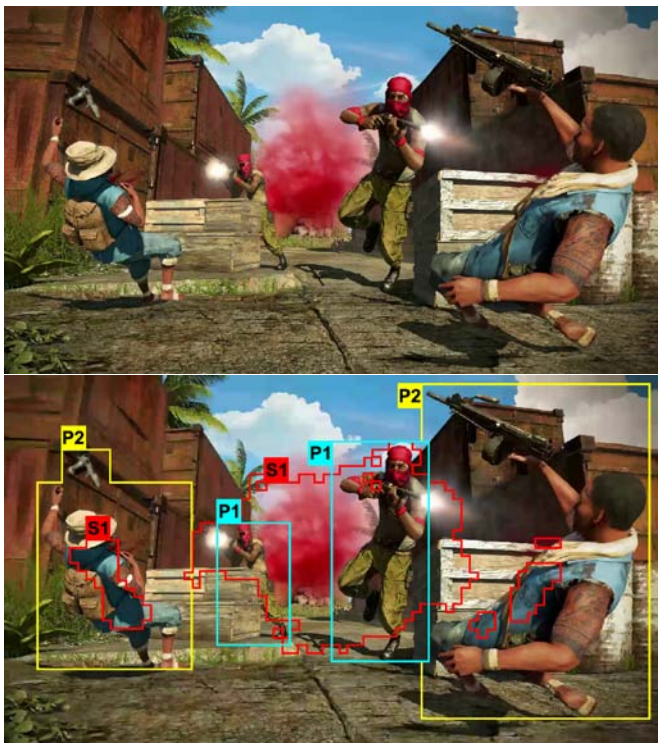

Fig. 5. A sample game frame encoded once by a single QP value (top) and another time with three QP values (bottom)

VI. CONCLUSION

In this paper, we introduced a conceptual Game Attention Model which determines the importance level of different regions of game frames according to user's attention. We then proposed an instance of such a model and showed that using this model would result in nearly 50 percent bit rate reduction on average. Having this model as a clue, we set the quantization parameter of each macro-block according to its importance level and hence made a balance between rate and distortion. Subjective quality assessment showed that Game Attention Model helps to decrease the bit rate while maintaining the user's quality of experience. This model would be beneficial in scalable coding and bit rate control in cloud gaming applications. Our future work is to investigate these possibilities and to design a rate control model that can match the bitrate of a cloud gaming session to the available and dynamically varying bandwidth of a player.

ACKNOWLEDGMENT

We would like to give our special thanks to those who kindly participated in our subjective assessment. We also appreciate members of MPL and DISCOVER Labs for their helpful comments.

REFERENCES

[1] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "An Evaluation of QoE in Cloud Gaming Based on Subjective Tests," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, 2011, pp. 330-335.

[2] K. T. Chen, Y. C. Chang, P. H. Tseng, C. Y. Huang, and C. L. Lei, "Measuring the latency of cloud gaming systems," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 1269-1272.

[3] S. Shi, C.-H. Hsu, K. Nahrstedt, and R. Campbell, "Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming," presented at the Proceedings of the 19th ACM international conference on Multimedia, Scottsdale, Arizona, USA, 2011.

[4] (15 Dec 2012). *NVIDIA Unveils Cloud GPU Technologies, Redefining Computing Industry For Third Time* [Online]. Available: http://nvidianews.nvidia.com/Releases/NVIDIA-Unveils-Cloud-GPU-Technologies-Redefining-Computing-Industry-for-Third-Time-7e2.aspx

[5] (15 Dec 2012). *OnLive* [Online]. Available: http://www.onlive.com

[6] M. Claypool, D. Finkel, A. Grant, and M. Solano, "Thin to win? Network performance analysis of the OnLive thin client game system," in *Network and Systems Support for Games (NetGames), 2012 11th Annual Workshop on*, 2012, pp. 1-6.

[7] L. Itti and C. Koch, "Computational modeling of visual attention," *Nature reviews neuroscience*, vol. 2, pp. 194-203, 2001.

[8] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *Computer Vision, 2009 IEEE 12th international conference on*, 2009, pp. 2106-2113.

[9] H. Rahimi, A. Nazari Shirehjini, and S. Shirmohammadi, "Activity-centric streaming of virtual environments and games to mobile devices," in *Haptic Audio Visual Environments and Games (HAVE), 2011 IEEE International Workshop on*, 2011, pp. 45-50.

[10] Y. Wang, H. Li, X. Fan, and C. W. Chen, "An attention based spatial adaptation scheme for H. 264 videos on mobiles," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 20, pp. 565-584, 2006.

[11] (15 Dec 2012). *H.264/AVC Reference Software* [Online]. Available: http://iphome.hhi.de/suehring/tml