A GAME ATTENTION MODEL FOR EFFICIENT BITRATE ALLOCATION IN CLOUD GAMING

Hamed Ahmadi[a], Saman Zad Tootaghaj[a], Mahmoud Reza Hashemi[a], Shervin Shirmohammadi[b]

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

[b] Distributed and Collaborative Virtual Environments Research Laboratory (DISCOVER Lab), School of Information Technology and Engineering, University of Ottawa, Canada, shervin@discover.uottawa.ca

Keywords: Cloud Gaming, Visual Attention Model, Bit Rate Reduction, Video encoding

ABSTRACT

The widespread availability of broadband internet access and the growth in server-based processing has provided an opportunity to run games away from the player into the cloud and offer a new promising service known as cloud gaming. The concept of cloud gaming is to render a game in the cloud and stream the resulting game scenes to the player as a video sequence over a broadband connection. In order to meet the stringent network bandwidth requirements of cloud gaming and support more players, efficient bitrate reduction techniques are needed. In this paper, we introduce the concept of Game Attention Model (GAM), which is basically a game context based 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 the less important macro-blocks with lower bit-rate.

We have evaluated nine game video sequences, covering a wide range of game genre and a spectrum of scene content in terms of details, motion and brightness. Our subjective assessment shows that by integrating this model into the cloud gaming framework, it is possible to decrease the required bit rate by nearly 25 percent on average, while maintaining a relatively high user quality of experience. This clearly enables players with limited communication resources to benefit from cloud gaming with acceptable quality.

1. 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 [1-5] 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 [6]. OnLive [7] and Marvell [8] announced a partnership to bring the OnLive cloud gaming platform to Google TV devices.

The concept of cloud gaming is to render a video game in the cloud and stream the game scenes as a video to game players over a broadband network. A cloud gaming framework is illustrated in Figure 1. In this framework, the user input signals (mouse, keyboard, or game controller events) are sent to the cloud to interact with the game application. As the Game Engine receives commands from the Game Controller Device, it updates the game state and renders the next game frame. This game frame is then encoded and sent over the network to the client's device where the received game frame is decoded and shown on its display. Cloud gaming has many advantages for users as well as game developers. On the 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 run 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.

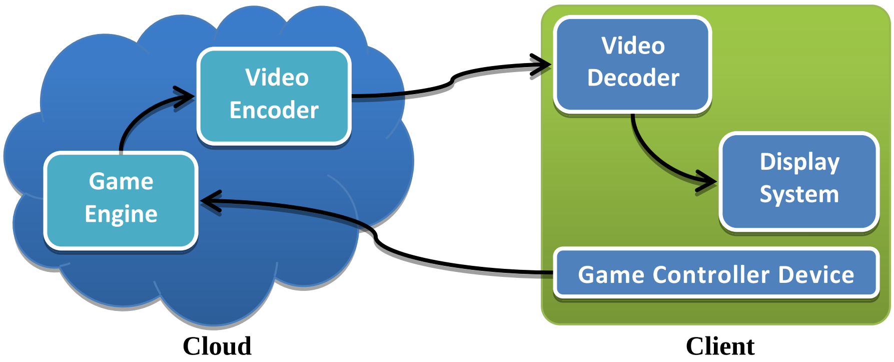

Figure 1. Cloud gaming concept diagram.

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 [4]. For example, OnLive 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 users who have lower bandwidth connectivity, such as mobile users. 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 players with limited computational and communication resources can still benefit from cloud gaming with acceptable quality. In terms of complexity, our model proposes some moderate load on the cloud side, because of its computationally light components and independence of its main blocks that can run in parallel. Besides, unlike a player's bandwidth, the cloud side is scalable, so supporting some additional complexity at the cloud side is justified by the benefit of being able to support more players who have lower bandwidths. As mentioned above, cloud gaming today requires very demanding bandwidths from players [4], so any technique that lowers that requirement is of great interest to the gaming industry.

Currently, Onlive and other cloud gaming companies use H.264/AVC as their basic video encoding standard except for Otoy [9] which owns a proprietary codec. However, regardless of how efficient the encoder is, the bandwidth of the streamed video is still higher than what most mobile users can afford. That's why most cloud gaming service providers look for ways to reduce this high bitrate even further. In this paper, we propose a new framework to achieve this goal. In our proposed framework, for a given game frame, each macro-block is encoded according to its "importance" from the player's perspective; i.e., if at a given moment a macro-block is less important for the player, it can be encoded with a lower quality/bitrate compared to more-important macro-blocks. Clearly, this approach can lead to a reduced video bitrate without significantly affecting the player's quality of experience. To achieve this goal, 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 for each of those regions. In this paper, we introduce a conceptual visual attention 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 [10]. 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 the neurobiological mechanisms behind human attention. 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, without loss of generality, we use Judd's Support Vector Machine (SVM) based model which has been trained using a large database of eye tracking data [11]. However, other similar models can be used as well and our methodology does not depend on any specific model.

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 users do not pay much attention to these less important regions maintains their overall quality of experience. Moreover, this approach implies that in some cases it is possible to choose an even lower QP value for important regions than that of a conventional encoding configuration and still have a lower or comparable bitrate. This can be done by choosing appropriately larger QP values for less important regions.

To summarize, in this paper we introduce a game context based visual attention model which takes game context into account and prioritizes regions of game frames from a game player's perspective. Our evaluations on nine representative game video sequences show that using our proposed model, it is possible to reduce the required bitrate of cloud gaming by nearly 25 percent in average, with a negligible loss in user quality of experience.

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

2. RELATED WORK

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.

Claypool et al. [4] provided a detailed study of the network characteristics of Onlive. They showed that cloud gaming appears most similar in downstream turbulence (considering bitrate, packet size and inter-packet time) to live video, requiring frequent transmission of large packets. This large turbulence suggests meeting the quality of service requirements of cloud gaming is a major challenge.

Chang et al. [12] proposed a methodology for measuring the performance of thin-clients on gaming. They concluded that display frame rate and frame distortion at the client side are both critical to gaming performance, where the frame rate is a much more important performance factor when designing a good thin-client for gaming. Their research validated a previous study [13] in which researchers quantified the effects of frame rate and resolution on user performance for conventional computer games. They had shown that frame rate has a marked impact on both player performance and game enjoyment while resolution has little impact on performance and some impact on enjoyment.

Since in cloud gaming, the game is streamed as video to the client, video adaptation techniques are applicable to this service. During the past decade, a vast literature on video adaptation has been generated, which offer a rich body of techniques for answering challenging questions. Such techniques usually transform the input video to an output video or augmented multimedia form by utilizing manipulations at multiple levels (signal, structural, or semantic) in order to meet diverse resource constraints and user preferences while optimizing the overall utility of the video [14]. A general conceptual framework to model video entity, adaptation, resource, utility, and relations among them can be found in [15]. This framework extends the conventional rate-distortion model in terms of flexibility and generality and allows for formulations of various adaptation problems as resource-constrained utility maximization. There have also been a number of researches for adapting video content for small displays of mobile devices. For example,[16] provides effective small size videos which emphasizes the important aspects of a scene while faithfully retaining the background context. This is achieved by explicitly separating the manipulation of different video objects.

Reducing the complexity and hence speeding up the video encoding process is another area of research. Authors in [17] present a system level complexity reduction for H.264/AVC video encoding by allocating resources based on computational complexity and quality trade-off. In their work, a framework is developed which allocates the computational power of the encoder according to video contents and also scales with the available battery power using an ROI classification method. Semsarzadeh et al. [18] proposed a joint multi rate optimization algorithm for H.264/AVC that generates an encoded bitstream that will have the minimum distortion across all the layers combined. Their proposed scheme not only supports bitrate adaption, but also is independent of the adaptation algorithm and is applicable to most existing adaptation frameworks. Recently, they presented the effect of encoder parameters on the encoding quality and power consumption using a Rate–Distortion–Complexity (R–D–C) model [19]. This model helps to effectively adapt to changes in the encoder’s software and hardware platforms, especially due to the power limitations of mobile devices.

Another kind of game streaming is geometry transmission, as opposed to video streaming, 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 for cloud gaming too. As an instance, in this paper, we use a graphics adaptation scheme for virtual environments using optimized object selection [20] such that only the most important objects from the perspective of the player's activity are included in the scene and less important objects are omitted.

Another branch of the research in the area of video content adaptation is focused on visual attention modeling. For example, [21] has proposed an attention-based spatial adaptation scheme which not only improves the perceptual quality but also saves the bandwidth and computation. [22] has used a visual attention model to predict, at a given video frame, how much attention viewers pay to each part of the frame. There are two kinds of attentions. We direct attention to objects in a scene using both bottom-up, image-based saliency cues and top-down, task-dependent cues. Attention allows us to break down the problem of understanding a visual scene into rapid series of computationally less demanding, localized visual analysis problems. In addition to these scene analysis functions, attention is also characterized by a feedback modulation of neural activity for the visual attributes and at the location of desired or selected targets [10]. Visual attention models try to computationally simulate these neurobiological mechanisms and measure the conspicuity of a location, or the likelihood of a location to attract the attention of human observers. They have already been used in many applications in graphics, design, and human computer interaction. For example, it can be used to direct foveated image and video compression [23,24] and levels of detail in non-photorealistic rendering [25]. It can also be used in advertising design, adaptive image display on small devices, or seam carving [26].

It is well-known that bottom-up visual attention models do not perform accurately, unless they are accompanied with top-down visual attention models. Current top-down visual attention models are not suitable for game context because they are usually designed to recognize an application specific pattern in videos. For example, visual attention models used for video conferencing applications are utilized with face detection algorithms. These models fail to estimate the attraction level of game videos due to various fantasy game characters. There are also some generic models which try to estimate the attraction level of different regions of videos by measuring the texture and motion activities[27]. Neither of these generic models are suitable for game context. For example, in many games there are moving objects in the background to which players hardly pay attention. These objects, however, are considered important with generic attention models due to their movement. Therefore, there is a need to design a game specific attention model to predict the importance level of different regions of game videos. This model would then be used in a video bitrate controller to properly adapt the encoder's parameters to meet the bandwidth constraint with the minimum possible reduction in users' perceived quality of experience.

Despite the huge literature on video content adaptation, practical methods which are suitable specifically for cloud gaming are rare [3] and there are still many open issues in this research area. 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.

3. DESIGN AND IMPLEMENTATION

3.1. DESIGN

In this paper, we introduce a model which helps the encoder to decrease the bit rate of game frames without noticeable loss in quality of experience. 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. Figure 2 shows where GAM fits in a generic cloud gaming framework.

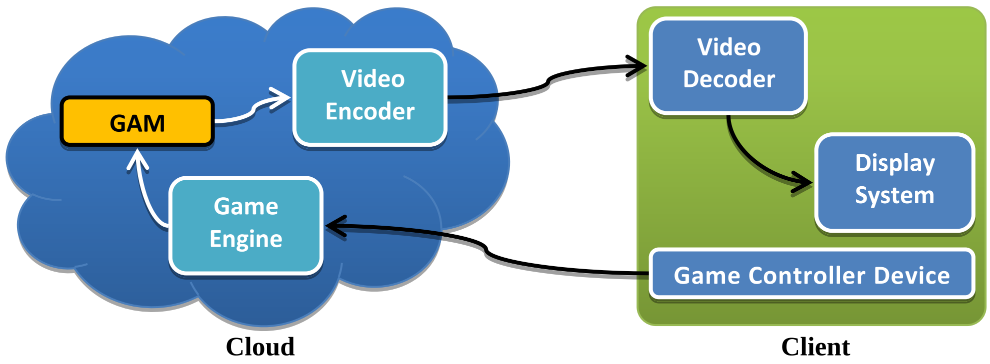

Figure 2. The position of GAM in a generic cloud gaming framework.

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 [11].

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.

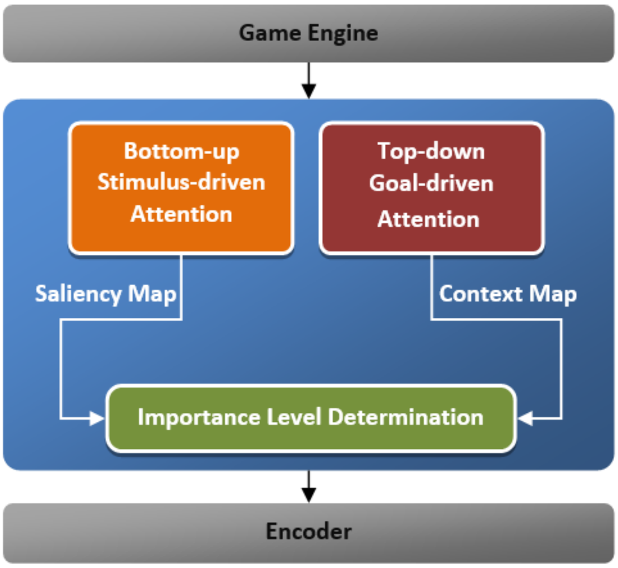

Figure 3. Proposed Game Attention Model.

As illustrated in Figure 3, we have considered both types of attentions in GAM. Each attention block produces a separate map. The map generated by the bottom-up attention block is called the "saliency map" and the one produced by the top-down attention block is called the "priority map". These maps need to be merged into a single map in which each pixel value indicates the importance level of that pixel. Importance Level Determination Block (ILDB)is responsible for this merging operation. Specifically, saliency and priority maps are fed into ILDB, where they are combined into a single map which is referred to in this paper as the "game attention map". ILDB assigns a value to each pixel of the game attention map based on the importance levels of corresponding pixels in the saliency and priority maps. For example, if a pixel were marked as important in both saliency and priority maps, ILDB would assign a high value to that pixel. ILDB can be implemented as easy as a pixel-wise maximum operation or as complex as an intelligent machine learning model. In addition to the merging operation, ILDB is also responsible to prepare the game attention map to be used in the encoder. Since the coding units of the H.264/AVC encoder are macro-blocks, ILDB divides the game attention map into blocks (of the same size of the encoder's macro-blocks) and assigns a value to each of them. Each block's value is assigned independently based on its inner pixel values. So, the importance level of a macro-block located in the i^{th} row and the j^{th} column ($MB_{i,j}$) is calculated by the following equation:

$$\textit{Importance Of } MB_{i,j} = \max(\textit{importance of all pixels in } MB_{i,j}) \tag{1}$$

Once the game attention map is ready, it is fed into the encoder. The encoder can then use this map to set the encoding parameters such as QP values such that important regions of the game frame are encoded with higher quality and those regions to which the user pay less attention are encoded with lower quality. This way we can reduce bitrate without noticeably decreasing the user's quality of experience. Note that the encoding parameters can be determined as easy as fixed values or as complex as values obtained through a rate control algorithm.

As we can see in Figure 3, GAM's inner blocks are interdependent, so these blocks can run in parallel. They can also be executed in a pipeline configuration. Figure 4 shows a diagram of such a pipeline. In this pipeline, six tasks have been defined: Frame Rendering, Axillary Frame Rendering, Priority Map Calculation, Saliency Map Calculation, Game Attention Map Calculation, and Frame Encoding. Note that the color of each block in GAM matches the color of that block's corresponding task in the pipeline. Frame Rendering is the same task that any game engine does. Axillary Frame Rendering is the process of rendering an extra frame which helps to generate the priority map. Priority Map Calculation task is responsible to calculate the priorities of objects in the current fame and reassign the colors of objects in the axillary frame based on their importance level. Saliency Map Calculation starts as soon as the current frame has been rendered. By the time the saliency map is ready, so is the priority map, thus the task of merging these two maps into a single game attention map can be done by the ILDB. In the Frame Encoding task, the rendered frame is encoded according to the generated game attention map.

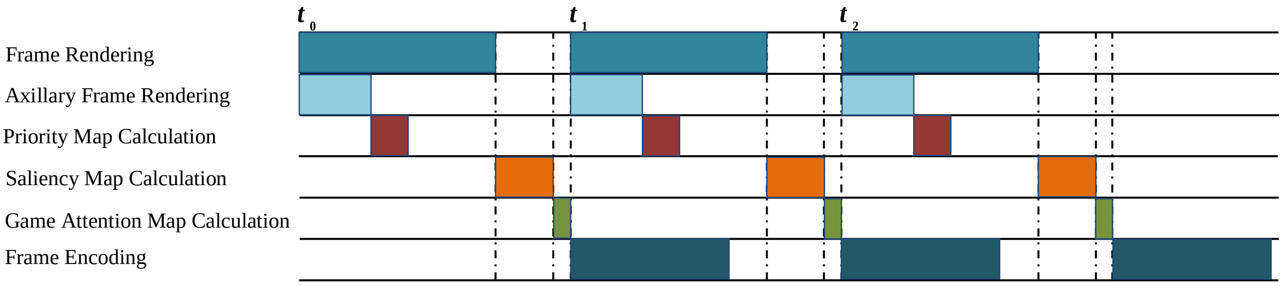

Figure 4. GAM’s execution in a pipeline configuration (not to scale).

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 3). Although this might not be the optimal selection, it adequately serves the purpose of this paper. This sample implementation is explained in more details in the next subsection.

3.2. 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 3. 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 [11], and for the top-down attention block, we opted the game priority model proposed in [20]. The regions highlighted by these two models do not necessarily overlap. 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.

GAM calculates a multi-level importance map from its input game frame. When the game engine starts rendering the current frame, it provides GAM with a list of current objects and player’s activity. It also starts rendering an extra frame in parallel. This extra frame contains the very same objects as the original frame does, but without any graphical effect. Each game object in this extra frame is assigned a unique color such that a 2D projected image of the game object on the frame could be distinguishable for further manipulations. Note that this extra rendering does not include any texture, shading or light processing, so it can be done even before the main frame has been rendered completely. In order to keep computational costs down, bounding-boxes of game objects can also be used. Meanwhile, the top-down attention block. assigns an importance factor to each game object, according to user’s current activity, as described in [20]. This factor is defined in several aspects of gameplay such as immersion, visual quality, task accomplishment and orientation. When the extra frame rendering is finished, it is fed into the top-down attention block. In this block, each game object in the frame which is now identifiable by its unique color is assigned a gray level according to its priority rank. In this paper, we have only considered three priority levels of high, medium, and low priority. Once the game engine has finished the rendering of the current game frame, it feeds it into GAM. In GAM, the bottom-up attention model calculates the saliency map. 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 adversely affect the performance of the model. In our work, this threshold has been determined experimentally so that the map contains proportionate amount of salient regions. This threshold can be used for the whole video sequence and it is not necessary to be set for each single game frame. It was set to seventy percent for all our evaluation video sequences. Lower values resulted in too large salient regions which were not helpful for reducing the bitrates of the video sequences. 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. In the H.264/AVC encoder the smallest unit on which the encoder operates is a macro-block, hence 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 have considered two weighting factors, so we 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. List 1 contains the pseudo-code of GAM.

Pseudo-code of Game Attention Model

```
// Suppose that the width and height of the video is 16*N and 16*M, respectively.

Input: frame_main: current rendered frame
Input: frame_axil: axillary rendered frame
Input: activity: current user activity
Input: objects []: list of objects in current frame
Input: cutoff: threshold for saliency map
Output: levels [,]: an N x M array containing the importance level of each macro-block

// Priority Block
priority_indexes ← getImportanceFactors(activity, objects);
normalized_priority_indexes ← getNormalized(priority_indexes);
prioritized_objects ← sortObjects(objects, priority_indexes);
map_priority ← getPriorityMap(frame_axil, prioritized_objects);

// Saliency Block
map_saliency_original ← getSaliencyMap(frame_main);
map_saliency ← applyThresholdOnSaliencyMap(map_saliency_original);

// Importance Level Determination Block
map_game_attention ← mergeMaps(map_saliency, map_priority);
levels ← getMacroblocksImportance(map_game_attention);
```

List 1. Pseudo-code of GAM.

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. As mentioned before, for simplicity and without lack of generality, in this paper we consider setting the quantization parameter QP. Hence, we 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 5 shows the output map of each block and the final attention map for three sample frames which are extracted from the video sequences we used for our evaluations. Five rows are seen in this picture. The first row (row A) includes the original game frames. The second row (row B) includes the saliency maps. These maps have been generated with Judd's attention model. The third row (row C) includes the saliency map after applying a threshold on them. This gives us a saliency map with two levels of importance. The fourth row (row D) includes the priority maps. These maps have been generated according to Rahimi's object prioritization scheme. These maps have three levels of importance. White, gray and black colors in these maps represent the high, medium and low priority regions, respectively. The last row (row E) includes the final game attention maps. The maps in rows C and D are merged into a single game attention map by ILDB.

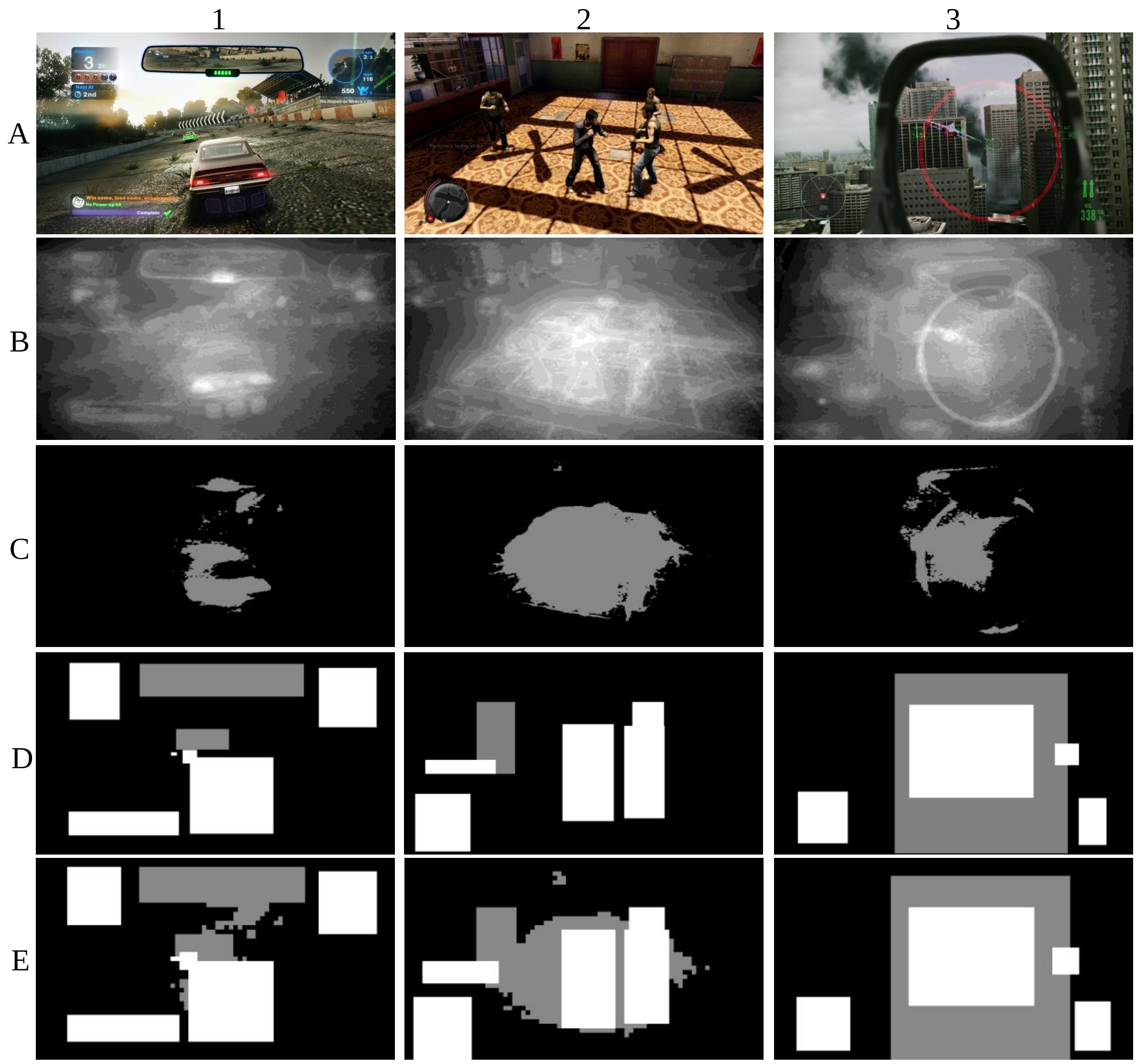

Figure 5. Results of each block in Figure 3 for three sample game frames. A) Game frames, B) Saliency map, C) Saliency map after applying the threshold, D) Priority map and E) GAM attention map.

The importance factors were set according to the guidelines in [20] by an expert game player. He specified the player's activities in each video sequence and assigned importance factors to the game objects according to each activity. All these importance factors have been summarized in Table 1. In this table, we categorized all visible objects in our evaluation video sequences into five groups: On-screen Information, Game Objects, Rivals, Team Players, and Surrounding Environment. On-screen Information group includes objects such as road maps, game hints and menus. Collectable and/or destroyable items were categorized into Game Objects group. Rivals are human or AI-enabled entities who try to win against or kill the player. Team Players group includes the player himself and his allies in game. All other objects which were not categorized in the latter four groups were listed in the Surrounding Environment group. For each user's activity in each video sequence, all of the objects in each group were assigned the same importance factor. Since we chose to have three levels of importance for priority maps, we asked our expert game player to assign importance factors with 0, 0.5 and 1.0 values to low, medium and high importance objects, respectively.

Table 1. Important factors for video sequences we used in our evaluations

Video Sequence Name	User's activity	On-screen Information	Game Objects	Rivals	Team Players	Surrounding Environment
COD	Shooting	1	0.5	1	0.5	0
	Exploring	1	1	1	0.5	0
SGW	Shooting	1	0.5	1	0.5	0
	Exploring	1	1	1	0.5	0
SLD	Fighting	1	0.5	1	0.5	0
BLR	Racing	0.5	1	1	1	0
NFS	Racing	0.5	1	1	1	0
PES	Playing	0.5	0.5	1	1	0
SNE1	Aiming	1	0.5	1	0.5	0
SNE2	Shooting	1	0.5	1	0.5	0
ACA	Fighting	1	0.5	1	0.5	0

4. Evaluation Results

Quality of experience (QoE) is a subjective measure of a customer's experiences with a service. In cloud gaming service, the quality of experience is required to be measured based on both game and video. Game-based experiences can be measured from different aspects such as immersion, tension and competence [28]. The quality of the streamed game video can affect these measures. For instance, video distortions can make players nervous (tension) or distract their fantasy (immersion). Low quality game videos can also affect the success of players in the game (competence). Distorted objects in game videos might result in the player missing an enemy or a vital sign in the game. For example, in a car racing game, if a player's car broke down and quality of the streamed game video for the player was not enough to find a repair sign in time, he would lose the race. So, video quality is among the factors that affect quality of user's experience in cloud gaming. In this paper, we conducted our evaluations on a set of game videos and measured the video quality. In our future work, we will integrate our proposed model in a practical cloud gaming test-bed. This way, it will be possible to directly measure immersion, tension and competence by means of a standard game experience questionnaire like the one proposed in [29].

Increasing 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 the player's current gaming context and viewing experience. In order to verify this assumption, we conducted subjective assessments. First, we gathered nine game video sequences from eight recent popular video games. Choosing these video sequences, we made sure they included enough details such as environmental objects, enemies and allies. These games are of different metrics such as genre, pace and luminance. Table 2 shows the information of the selected games.

Table 2. Information of the selected games

Game	Abbreviation	Genre	Release
CALL OF DUTY BLACK OPS II	COD	First Person Shooter	2012
SNIPER GHOST WARRIOR 2	SGW	Tactical Shooter	2013
SLEEPING DOGS	SLD	Action-Adventure	2012
BLUR	BLR	Racing, Vehicular Combat	2010
NEED FOR SPEED THE RUN	NFS	Racing	2011
PRO EVOLUTION SOCCER 2013	PES	Sports	2012
SNIPER ELITE V2	SNE1/SNE2	Tactical Shooter, Stealth	2012
ACE COMBAT: ASSAULT HORIZON	ACA	Arcade, Combat Flight Simulator	2011

In the COD video sequence, the main character is in a fight with an enemy in a room. There are three doors in front of him from which enemies could appear. Among other things seen in the room are a couple of tables and some wooden cubes. This is a low motion, low content, and dark scene.

In the SGW video sequence, the player is in a dense jungle and kills the guards of a bridge. There are bushes, trees, grass and stones in the scene. SGW contains dark, moderately detailed and relatively fast motion content.

In the SLD video sequence, the main character is in a room which has been decorated in Chinese style. There are two enemies in the room and the player engages in a two-on-on fight with them. A friend of the main character is also seen in the scene. This is a moderately high motion, with moderate details, and bright scene.

In the BLR video sequence, the player is in a car race. Two other cars are seen in the scene. There are some power-ups that the players can grab and use to attack others or shield his car from the others' attacks. This is a high motion, moderately high content, and moderately bright scene.

In the NFS video sequence, the main character escapes in his car while two other cars are chasing him. The race is taking place in streets of a town. There are flashing direction signs on sides of the streets which help the player react in time to get past sharp turns. The map of the town is also partially seen on the screen. This is a high motion, moderate content, and moderately dark scene.

In the PES video sequence, the player carries the ball behind the opponent's box, passes the ball to the forward and shoots towards the goal. Meanwhile, the opponent's players are trying to get the ball back. This is moderately high motion since, despite the details (the players) that it should preserve, it has large areas of plain background (the grass). It is also a bright scene.

We recorded two video sequences from SNE. In the SNE1 video sequence, the main character acts as a sniper. He stands in a higher position and zooms his sniper on one of the two enemies that are talking to each other in a distance. This has dark, low motion and low content areas around the main scene, as well as bright, high content but low motion regions in the center.

In the SNE2 video sequence, the player is hiding behind a statue and tries to kill the enemy soldier in front of him. They are in the yard of a building and there is a car next to them which is on fire. This has moderate motion, high content, and bright scene.

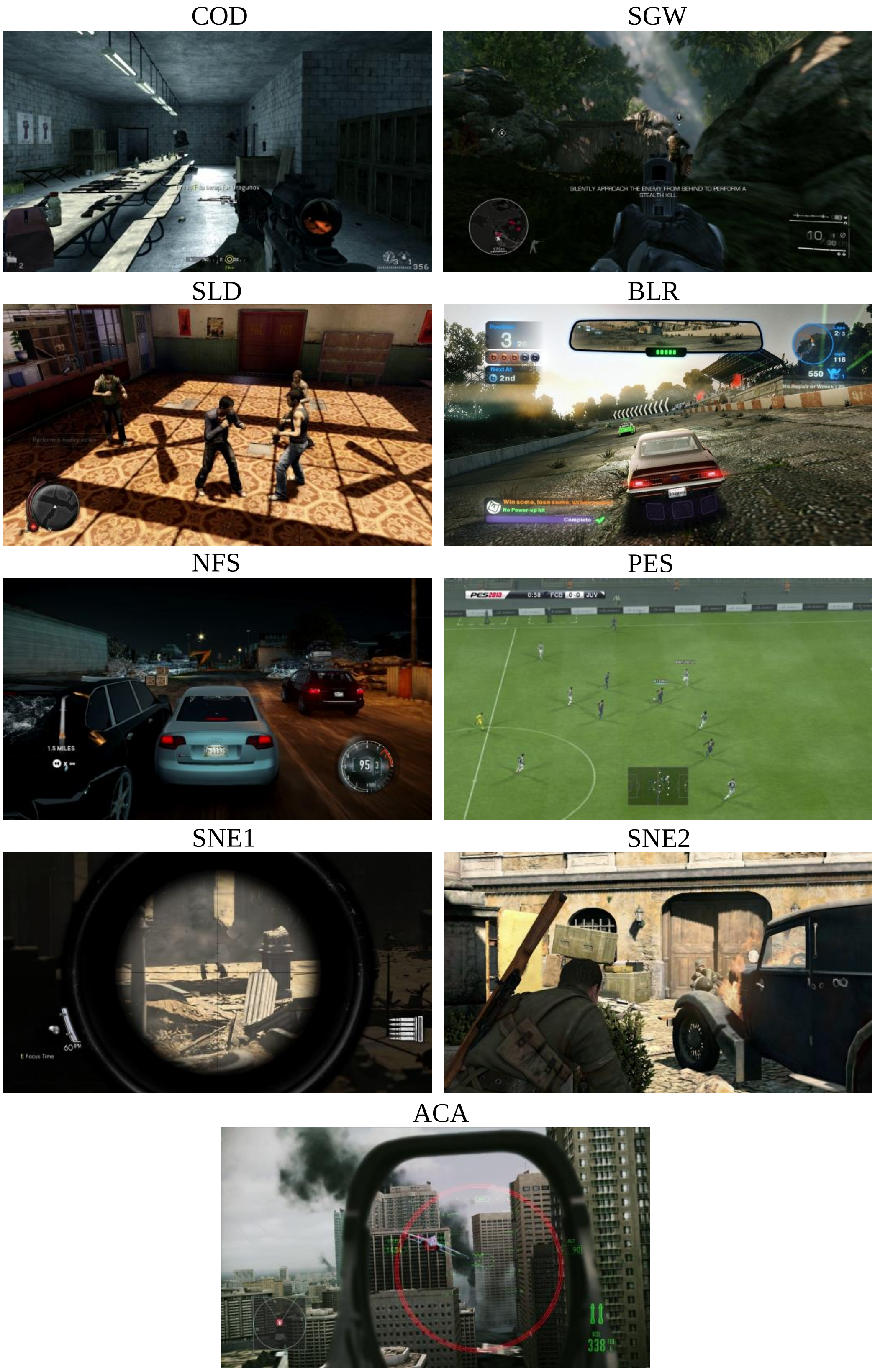

Figure 6. The first frame of each of the nine video sequences that we used in our evaluations.

Finally, in the ACA video sequence, the player tries to take the enemy aircraft down. They are flying on top of a city with high buildings. A head-up display is also seen on the screen. This is a high motion, high content, and bright scene.

The selected sequences above cover a wide range of game genres and contain a variety of motion content and spectrum of scene content in terms of details and brightness. Hence, the evaluation results will show the potentials of the proposed GAM concept for a wide variety of games.

Figure 6 shows the first frame of each of the nine video sequences we used in our evaluations. All sequences were recorded at 30 fps and saved with a lossless codec using a commercial application. Each video sequence lasts 4 seconds, and consists of 120 frames.

To encode our videos, we used the JM V18.4 software [30], which is an implementation of H.264/AVC used in many current cloud gaming companies. 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.

We encoded each video sequence 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 video sequences. In all cases, we chose a GOP size of 15, including an I-frame followed by fourteen P-frames. Table 3 summarizes the encoding parameters that we used in our evaluations.

Table 3. The name and value of H.264/AVC encoding parameters that we used in our evaluations

Parameter Name	Parameter Value
Profile	Baseline
Level	3.1
Number of Reference Frames	1
Motion Estimation Scheme	EPZS
Search Range	32
RD-optimized mode decision	ON
Rate Control	OFF

The reference game video sequences in our experiment were encoded without GAM and hence using a single QP value. In order for the comparison to be fair, we asked an expert to choose the maximum QP value for which he could not detect any distortion in these frames. To do so, we started with a QP value of 26 and increased it by one, step by step. At each step, we asked our expert whether he had noticed any distortions. Then we chose the maximum QP value for which the expert did not report any distortions. This maximum QP value was 30 and the average PSNR of the references for this QP value was 36 dB. Figure 7 shows how PSNR decreases when QP value increases for the COD video sequence.

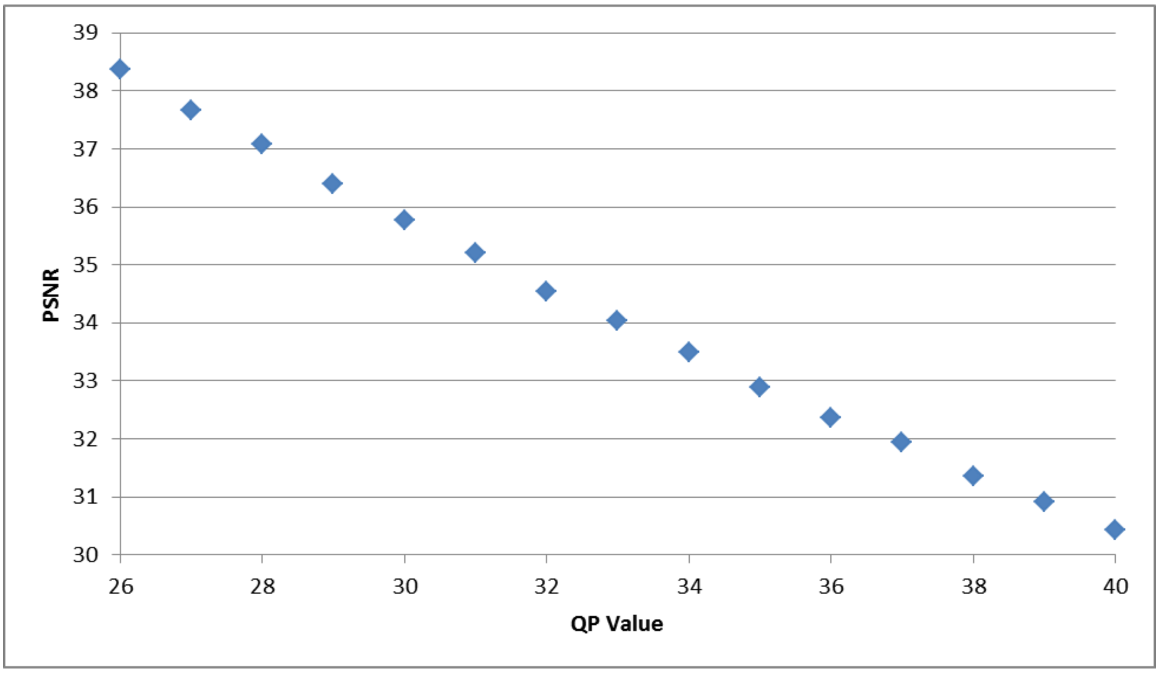

Figure 7. PSNR of COD video sequence encoded with different QP values.

The reference video sequences were once again encoded with the help of GAM. In this case, each sequence 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. In order to choose an appropriate set of QP values we investigated all possible triples of the form (30 + 2X, 30 + X, 30) where 1 <= X <= 5. Figure 8 illustrates the rate-distortion diagram of these choices for the COD sequence. In this figure, as X increases, both bitrate and PSNR decrease. However, our subjective evaluation shows that some decrease in PSNR is tolerable and does not affect players' perceived quality because the QP value for the important regions were kept as small as that of the single QP scenario.

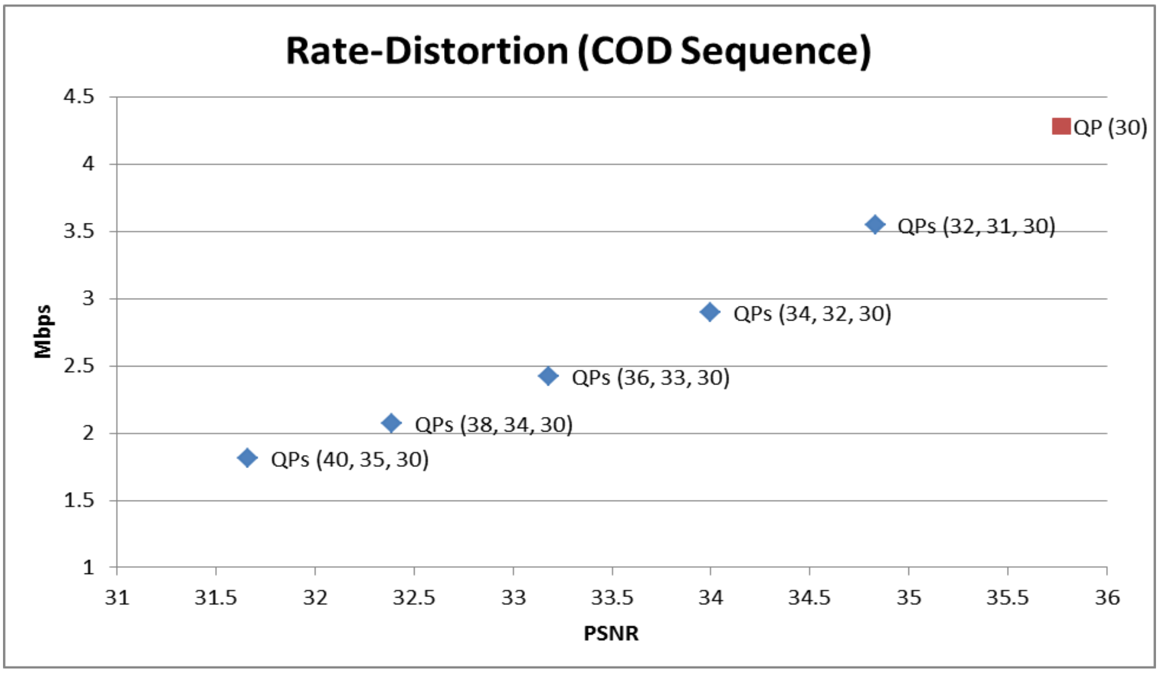

Figure 8. Rate-distortion diagram of COD video sequence encoded with different QP values.

Table 4 shows PSNR, SSIM and bitrate of the all nine video sequences when they were encoded once with a single QP value of 30 and other times with the QP sets mentioned above.

Table 4. PSNR, SSIM and bitrate of the video sequences for different QP values (PSNR and bitrate have been reported in dB and Mbps, respectively)

	COD			SGW			SLD		
Scenario	PSNR	SSIM	Bitrate	PSNR	SSIM	Bitrate	PSNR	SSIM	Bitrate
QP (30)	35.77	0.89	4.28	36.4	0.91	3.47	35.02	0.91	6.15
QPs (32, 31, 30)	34.83	0.87	3.54	35.44	0.89	2.67	34.13	0.89	5.28
QPs (34, 32, 30)	34	0.85	2.89	34.58	0.88	2.08	33.33	0.87	4.65
QPs (36, 33, 30)	33.18	0.83	2.41	33.71	0.86	1.66	32.50	0.85	4.21
QPs (38, 34, 30)	32.39	0.82	2.07	32.88	0.85	1.40	31.67	0.83	3.92
QPs (40, 35, 30)	31.66	0.8	1.80	32.11	0.83	1.22	30.91	0.81	3.71
	BLR			NFS			PES		
Scenario	PSNR	SSIM	Bitrate	PSNR	SSIM	Bitrate	PSNR	SSIM	Bitrate
QP (30)	36.76	0.93	12.81	39.89	0.97	4.52	39.44	0.94	1.05
QPs (32, 31, 30)	35.85	0.92	11.16	39.21	0.96	4.04	38.8	0.93	0.99
QPs (34, 32, 30)	34.96	0.91	9.77	38.56	0.96	3.64	38.16	0.93	0.87
QPs (36, 33, 30)	33.93	0.9	8.53	37.87	0.95	3.32	37.47	0.93	0.77
QPs (38, 34, 30)	32.91	0.89	7.57	37.07	0.95	3.07	36.68	0.92	0.70
QPs (40, 35, 30)	31.92	0.88	6.80	36.37	0.94	2.87	35.86	0.92	0.64
	SNE1			SNE2			ACA		
Scenario	PSNR	SSIM	Bitrate	PSNR	SSIM	Bitrate	PSNR	SSIM	Bitrate
QP (30)	39.85	0.97	1.77	36.8	0.94	5.37	39.02	0.97	6.23
QPs (32, 31, 30)	39.05	0.97	1.51	35.91	0.93	4.58	38.26	0.97	5.72
QPs (34, 32, 30)	38.27	0.96	1.25	35.02	0.92	3.77	37.47	0.96	5.17
QPs (36, 33, 30)	37.52	0.96	1.06	34.14	0.91	3.19	36.65	0.96	4.74
QPs (38, 34, 30)	36.69	0.95	0.91	33.24	0.90	2.73	35.72	0.95	4.37
QPs (40, 35, 30)	35.91	0.95	0.77	32.34	0.88	2.32	34.77	0.94	4.04

Table 5. Bitrate reduction percentage of multi-QP scenarios

	COD	SGW	SLD	BLR	NFS	PES	SNE1	SNE2	ACA	Average
QPs (32, 31, 30)	17.10	22.77	14.07	12.89	10.52	6.11	14.45	14.70	8.14	13.42%
QPs (34, 32, 30)	32.34	39.83	24.34	23.71	19.32	16.93	29.35	29.66	17.08	25.84%
QPs (36, 33, 30)	43.46	51.9	31.60	33.44	26.41	26.3	40.10	40.56	23.94	35.30%
QPs (38, 34, 30)	51.54	59.49	36.24	40.87	31.91	33.31	48.54	49.15	29.89	42.33%
QPs (40, 35, 30)	57.72	64.59	39.65	46.92	36.48	38.91	56.21	56.67	35.11	48.03%

Table 5 shows the bitrate reduction percentage that is achieved by using multiple QP values rather than using single value of 30. According to this table, although it is possible to decrease the bitrate up to 50 percent on average by choosing QP set (40, 35, 30), we did not choose this set for our subjective assessment because distortion in low priority regions were noticeable enough to affect the user's quality of experience. Rather we chose the QP set that corresponds to the just-noticeable difference. For these sequences this happens to be 34, 32 and 30 for the low, medium and high priority macro-blocks, respectively.

By using FMO, we might have blockiness either within or between slices. By choosing QP value of 30 for important regions, we made sure blockiness was avoided in these regions. As mentioned above, by setting QP value to 30, the average PSNR of the videos which we used in our subjective evaluation is 36dB. Higher QP values (32 and 34 for medium and low priority regions, respectively) are more likely to produce blockiness. In fact, the more texture and/or motion activity a video sequence has, the more likely blockiness is noticeable. For example, in the PES video sequence which has high motion activity, blockiness is more noticeable. This is also true for the SNE2 video sequence in which the car on fire has high motion and texture activity. But the point is that players pay little attention to low priority regions of game videos; in fact this is the main rationale behind our proposed approach and why the approach works. If there were some blockiness in unimportant regions, players would hardly notice.

In the multi-QP scenario, the quantization steps need to be chosen close to each other to avoid blockiness. In this scenario, blockiness might be noticeable in borders of slices. In our evaluation, we have chosen 30, 32 and 34 as QP values of high, medium and low priority regions, respectively and subjective evaluation shows that this set of values does not introduce noticeable blockiness that would affect the gaming experience.

To further evaluate the impact of each of the two attention models, we encoded each video sequence once with only the priority map, with the same three QPs, and another time with only the saliency map, 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 method described next.

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). Twenty viewers participated in this experiment. All of them were non-experts with no expertise in video processing and image quality assessment. Table 6 shows the demographic profile of the subjects.

Table 6. Demographics of the subjects

Gaming Experience				
Bad	**Poor**	**Good**	**Fair**	**Excellent**
25%	15%	35%	15%	10%
Monthly Game Play				
<= 5	**6 - 10**	**11 - 20**	**21 - 30**	**> 30**
35%	20%	30%	10%	5%
Gaming Platform (Already Played on)				
PC	**Console**	**Tablet**	**Cellphone**	
80%	35%	5%	0%	
Genres (Already Played)				
First Person Shooter	**Adventure**	**Sports**		
85%	35%	80%		
Racing	**Tactical Shooter**	**Flight Simulator**		
95%	20%	10%		

The video sequences in our study were viewed by each subject and required twenty minutes of their time in total. The sequences 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-second 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. 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 opinion.

Since the standard allows us to define the quality for the subjects, we used the same definitions as in [31] to measure the user experience. These definitions are reported in Table 7.

Table 7. Ratings and their definitions

Ratings	Definition
4.5—5.0	Excellent game, no impairment at all
4.0—4.5	Minor impairment, will not quit game
3.0—4.0	Impairment noticeable, might quit the game
2.0—3.0	Clearly impairment, usually quit the game
1.0—2.0	Annoying environment, definitely quit

Table 8 presents the results of our experiment for two tested scenarios. As mentioned before, in one scenario we used only the priority map in GAM 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 (priority only). In Table 8, the subjective quality for each scenario is expressed as the mean opinion score (MOS). In this table, the bit rate (KB), PSNR (dB) and SSIM index for the game video sequences in our experiment have also been included. The last column of Table 8 compares each of the two multi QP scenarios with the single QP scenario in terms of percentage of reduction in the subjective/objective metrics.

Table 8. Results of our experiment for the two scenarios

	Mean Opinion Score									
	COD	SGW	SLD	BLR	NFS	PES	SNE1	SNE2	ACA	Change
Single QP	5	4.87	5	5	5	4.82	5	4.92	4.97	-
Multi QP (Priority Only)	4.47	4.57	4.92	4.8	4.62	3.6	4.45	4.6	4.55	-9.02%
Multi QP (Saliency + Priority)	4.44	4.65	5	4.95	4.72	3.8	4.55	4.6	4.7	-7.15%
	Bitrate (Mbps)									
	COD	SGW	SLD	BLR	NFS	PES	SNE1	SNE2	ACA	Change
Single QP	4.28	3.47	6.15	12.81	4.52	1.05	1.77	5.37	6.23	-
Multi QP (Priority Only)	2.78	2.00	4.5	9.63	3.53	0.82	1.01	3.57	5.15	-29.62%
Multi QP (Saliency + Priority)	2.89	2.08	4.65	9.77	3.64	0.87	1.25	3.77	5.17	-25.84%
	PSNR (dB)									
	COD	SGW	SLD	BLR	NFS	PES	SNE1	SNE2	ACA	Change
Single QP	35.77	36.4	35.02	36.76	39.89	39.44	39.85	36.8	39.02	-
Multi QP (Priority Only)	33.9	34.48	33.19	34.85	38.36	38.13	37.68	34.84	37.46	-4.76%
Multi QP (Saliency + Priority)	34	34.58	33.33	34.96	38.56	38.16	38.27	35.02	37.47	-4.33%
	SSIM									
	COD	SGW	SLD	BLR	NFS	PES	SNE1	SNE2	ACA	Change
Single QP	0.89	0.91	0.91	0.93	0.97	0.94	0.97	0.94	0.97	-
Multi QP (Priority Only)	0.85	0.88	0.87	0.91	0.96	0.93	0.96	0.92	0.96	-2.29%
Multi QP (Saliency + Priority)	0.85	0.88	0.87	0.91	0.96	0.93	0.96	0.92	0.96	-2.29%

As we can see from the results, GAM has achieved about 25 percent bit rate reduction on average. Obviously, such a reduction would decrease PSNR and SSIM as it has here. But the small actual perceived quality reduction of 7.15%, as reported by the subjects in their MOS is a testimony that this distortion has been mostly hidden to the viewers and has not significantly affected their quality of experience with the game. Note that according to the definitions of the ratings, the subjects noticed no impairment in seven out of nine videos.

The subjective quality is better in "Saliency + Priority" compared to the "Priority only" scenario. Although “Priority Only” achieves 29.62 – 25.84 = 3.78% more bitrate reduction than "Saliency + Priority", the latter improves the quality as perceived by the players by a factor of 9.02/7.15 = 1.26 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 9 shows the average percentage of low, medium and high priority regions in ‘Saliency’, ‘Priority Only’ and ‘Priority+Saliency’ attention maps.

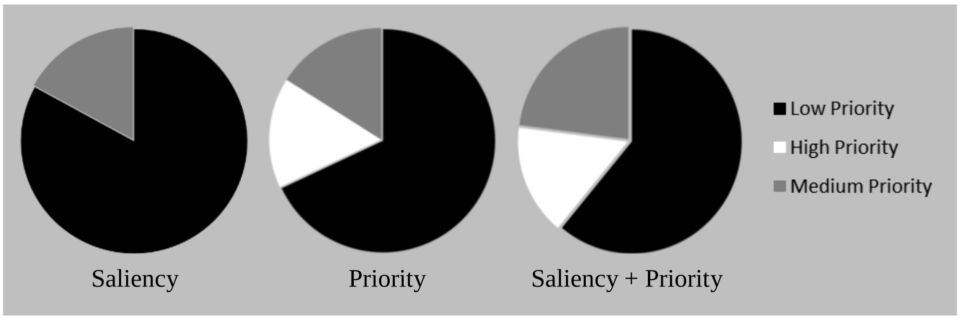

Figure 9. The average percentage of low, medium and high priority regions in ‘Saliency’, ‘Priority Only’ and ‘Priority+Saliency’ attention maps.

A

B

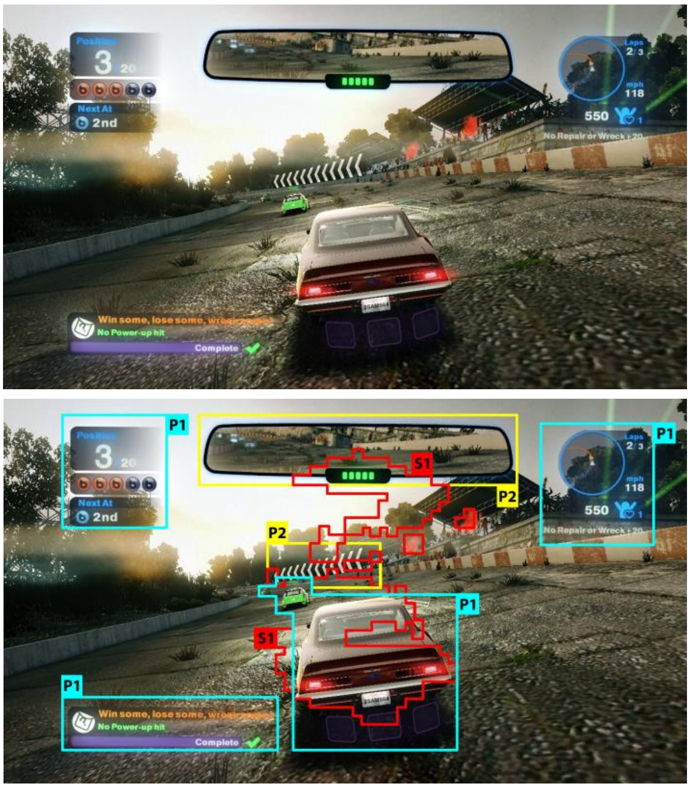

Figure 10. The first frame of BLR video sequence encoded once by a single QP value (A) and another time with three QP values (B).

5. DISCUSSION AND FUTURE WORK

Despite the enticing advantages of cloud gaming, high bandwidth requirements of this service limits its number of customers. Therefore, there is a need to develop efficient encoders to decrease bit rate while maintaining the quality. Quality of experience depends on genre of the game. For example, [1] showed fast games are more tolerant towards loss than others. In fast games, like First Person Shooters, since the player's success depends on how fast he reacts, he is less likely to pay attention to the details of graphic objects. In contrast, the player of medium-paced games, like Third Person Shooters, is more interested in what he sees (i.e. video quality).

Our proposed model can be utilized with a variety of video coding scenarios. For example, it is possible to build FMO on top of another error-resilience tool, Arbitrary Slice Ordering, because each slice group can be sent in any order and can optionally be decoded in order of receipt, instead of in the usual order of scan. In this configuration, our model would be of great help to efficiently group the macro-blocks.

As our future work, we plan to investigate the possibility of integrating GAM into an SNR-based scalable video coding scenario. A probable implementation would be to design the base layer to include the more informative transform coefficient and send the remaining coefficients through enhancement layers. The number of coefficients to be included in the base layer could possibly be determined by GAM. The more important the macro-block, the more coefficients of that macro-block would be transmitted in the base layer. Therefore, the bit rate of the base layer decreases while providing acceptable quality in case of losing enhancement layers.

Communication of the priority information between GAM and the encoder can be done by means of MPEG-7 descriptors. Therefore, it is possible to use an existing MPEG-7-enabled encoder without any modifications and keep the development costs down.

The goal of this paper is to show that it is possible to reduce (and in future control) the required bandwidth of cloud gaming by considering the features of Human Visual System (HVS). That is why we chose intuitive methods for GAM’s implementation. The subjective results prove that we have achieved this goal and now we can move further and enhance the model. This enhancement will include the bottom-up attention block of GAM, too. We hypothesize that region growing segmentation would be a feasible solution. However, it needs to be tested and analyzed in terms of accuracy and time complexity.

According to Table 8, the subjective quality of PES, SGW and SNE2 video sequences was reported to be less than that of other video sequences. When we asked the subjects to explain their ratings for these three video sequences, we realized that distortions in the regions with high texture or movement activities, even in regions of lower importance, would both distract and annoy players. For example, despite being less important, fire sparks in SNE2 had fast movements and did in fact grab the attention of the players, adversely affecting the subjective quality of the video sequence. Another example is the field of football match in PES. Although players hardly pay attention to the green parts of the field, distortions in these parts would irritate them. This is due to the near plane textures of these parts which make distortions more conspicuous.

A feasible solution to avoid this problem is to choose lower QP values for the encoding of GOPs with higher texture or movement activities. As our future work, we plan to model this relationship and adaptively set the QP value of each region according to the amount of its texture and movement activities.

The last issue to discuss is about our evaluation methodology. The methodology used here is originally designed for assessment of the quality of broadcasting services, and not interactive services like cloud gaming. Moreover, we asked the subjects' opinions about the game video sequences, while it would be interesting to get their opinion if they played the game. Thus, we are going to develop a suitable test bed as in [1] for our future research on cloud gaming and use a specific game experience questionnaire like [29].

6. CONCLUSIONS

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 25 percent bit rate reduction on average. According to the output of this model, 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.

ACKNOWLEDGEMENT

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. Jarschel M, Schlosser D, Scheuring S, Hoßfeld T 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. IEEE, pp 330-335
2. Chen KT, Chang YC, Tseng PH, Huang CY, Lei CL Measuring the latency of cloud gaming systems. In: Proceedings of the 19th ACM international conference on Multimedia, 2011. ACM, pp 1269-1272
3. Shi S, Hsu C-H, Nahrstedt K, Campbell R (2011) Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming. Paper presented at the Proceedings of the 19th ACM international conference on Multimedia, Scottsdale, Arizona, USA,
4. Claypool M, Finkel D, Grant A, Solano M 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. IEEE, pp 1-6
5. Hemmati M, Javadtalab A, Shirehjini AAN, Shirmohammadi S, Arici T (2013) Game as Video: Bit Rate Reduction through Adaptive Object Encoding. Paper presented at the NOSSDAV'13, Oslo, Norway, February 26-March 11, 2013
6. NVIDIA Unveils Cloud GPU Technologies, Redefining Computing Industry For Third Time. http://nvidianews.nvidia.com/Releases/NVIDIA-Unveils-Cloud-GPU-Technologies-Redefining-Computing-Industry-for-Third-Time-7e2.aspx. Accessed 15 Dec 2012
7. OnLive. http://www.onlive.com. Accessed 15 Dec 2012
8. Marvel. http://www.marvell.com/. Accessed 29 Mar 2013
9. Otoy. http://www.otoy.com. Accessed 31 Mar 2013
10. Itti L, Koch C (2001) Computational modeling of visual attention. Nature reviews neuroscience 2 (3):194-203
11. Judd T, Ehinger K, Durand F, Torralba A Learning to predict where humans look. In: Computer Vision, 2009 IEEE 12th international conference on, 2009. IEEE, pp 2106-2113

12. Chang Y-C, Tseng P-H, Chen K-T, Lei C-L Understanding the performance of thin-client gaming. In: Communications Quality and Reliability (CQR), 2011 IEEE International Workshop Technical Committee on, 2011. IEEE, pp 1-6
13. Claypool M, Claypool K, Damaa F The effects of frame rate and resolution on users playing First Person Shooter games. In: Electronic Imaging 2006, 2006. International Society for Optics and Photonics, pp 607101-607101-607111
14. Chang S-F, Vetro A (2005) Video adaptation: concepts, technologies, and open issues. Proceedings of the IEEE 93 (1):148-158
15. Kim J-G, Wang Y, Chang S-F Content-adaptive utility-based video adaptation. In: Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on, 2003. IEEE, pp III-281-284 vol. 283
16. Cheng W-H, Wang C-W, Wu J-L (2007) Video adaptation for small display based on content recomposition. Circuits and Systems for Video Technology, IEEE Transactions on 17 (1):43-58
17. Kannur AK, Li B Power-aware content-adaptive H. 264 video encoding. In: Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, 2009. IEEE, pp 925-928
18. Semsarzadeh M, Hashemi MR A joint multi rate optimization framework for video adaptation in H. 264/AVC. In: Consumer Electronics (ISCE), 2010 IEEE 14th International Symposium on, 2010. IEEE, pp 1-4
19. Semsarzadeh M, Lotfi A, Hashemi MR, Shirmohammadi S (2013) A fine-grain distortion and complexity aware parameter tuning model for the H. 264/AVC encoder. Signal Processing: Image Communication
20. Rahimi H, Nazari Shirehjini A, Shirmohammadi S 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. IEEE, pp 45-50
21. Wang Y, Li H, Fan X, Chen CW (2006) An attention based spatial adaptation scheme for H. 264 videos on mobiles. International Journal of Pattern Recognition and Artificial Intelligence 20 (04):565-584
22. Chen LQ, Xie X, Fan X, Ma WY, Zhang HJ, Zhou HQ (2003) A visual attention model for adapting images on small displays. Multimedia systems 9 (4):353-364
23. Wang Z, Lu L, Bovik AC (2003) Foveation scalable video coding with automatic fixation selection. Image Processing, IEEE Transactions on 12 (2):243-254
24. Geisler WS, Perry JS A real-time foveated multi-resolution system for low-bandwidth video communication. In: Proceedings of the SPIE: The International Society for Optical Engineering, 1998. pp 294-305
25. DeCarlo D, Santella A Stylization and abstraction of photographs. In: ACM Transactions on Graphics (TOG), 2002. ACM, pp 769-776
26. Rubinstein M, Shamir A, Avidan S Improved seam carving for video retargeting. In: ACM Transactions on Graphics (TOG), 2008. ACM, p 16
27. Shen L, Liu Z, Zhang Z (2013) A novel H. 264 rate control algorithm with consideration of visual attention. Multimedia Tools and Applications:1-19
28. Nacke LE, Drachen A, Kuikkaniemi K, Niesenhaus J, Korhonen HJ, Hoogen vdW, Poels K, IJsselsteijn W, Kort Y Playability and player experience research. In: Proceedings of DiGRA, 2009.
29. Poels K, IJsselsteijn W (2008) Development and validation of the Game Experience Questionnaire. Paper presented at the FUGA Workshop mini-symposium, Helsinki, Finland,
30. H.264/AVC Reference Software. http://iphome.hhi.de/suehring/tml. Accessed 15 Dec 2012
31. Wang S, Dey S Modeling and characterizing user experience in a cloud server based mobile gaming approach. In: Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, 2009. IEEE, pp 1-7