Game as Video: Bit Rate Reduction through Adaptive Object Encoding

Mahdi Hemmati[1], Abbas Javadtalab[1], Ali Asghar Nazari Shirehjini[1],
Shervin Shirmohammadi[1,2], Tarik Arici[2]

[1] Distributed and Collaborative Virtual Environments Research Laboratory (DISCOVER Lab)
School of Electrical Engineering & Computer Science, University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada

[2] Machine Learning Laboratory, Istanbul Sehir University, 34662 Istanbul, Turkey

mhemmati@eecs.uottawa.ca, javadtalab@discover.uottawa.ca, anazari@discover.uottawa.ca,
shervin@discover.uottawa.ca, tarikarici@sehir.edu.tr

ABSTRACT

Wide-spread availability of broadband internet access and the ubiquity of thin smart end devices such as smart-phones and tablets have led to a trend of moving more services away from the end devices to data centers, commonly referred to as *Cloud Computing*. For gaming, the stringent network bandwidth requirements of *Cloud Gaming* and the rapid growth of massively multiplayer online games (MMOG) call for novel content encoding and scene customization schemes to control the bit rate of the streaming video of the game scene. In this paper, a first step has been taken towards this goal by presenting a selective object encoding method to reduce the required network bandwidth and processing power without much impact on the player's quality of experience.

Categories and Subject Descriptors

I.3.8 [**Computer Graphics**]: Applications.

General Terms

Algorithms, Design.

Keywords

Cloud Gaming; Video Streaming; Content Adaptation; Object Selection and Optimization; Thin Clients.

1. INTRODUCTION

Cloud Gaming is a newly emerged gaming paradigm, which combines the successful concepts of *Cloud Computing* and *Online Gaming*. Cloud gaming, which could be defined as real-time game playing via thin clients [1], essentially moves the game logic required to run a game away from the user into a data center (the *Cloud*) and then streams the entire game experience to the user. The streaming can be done in three ways: streaming the 3D graphics and update messages to the player (which is the classical approach), encoding the game frames as video and streaming the video to the player (such as OnLive [2]), or a hybrid approach of streaming graphics primitives as well as video simultaneously (such as CiiNOW [3]). While each of these three methods has its own pros and cons, in this paper we focus on the video streaming paradigm since it is a new paradigm that has been commercially successful and is growing rapidly. Using video streaming, the player is no longer required to possess high-end gaming hardware.

The only requirement for the client side is broadband internet connection and the ability to display high quality video. We can also call this paradigm “Game as Video” (GaV). The client is only responsible for sending the control inputs of the player to the server, and displaying the received video of the gameplay. The entire process of rendering 3D objects and encoding the resulting raw video would be performed by the server(s) in the cloud.

Migrating from conventional console gaming or even online gaming to GaV offers a number of advantages for both game players and game developers. GaV frees players from the need to constantly upgrade their computers, as they can now play games hosted on remote servers with a broadband Internet connection and a thin client. Here a thin client can be a lightweight PC, a TV with a set-top box, a smartphone, a tablet, etc. Consequently, there are no setup overheads or compatibility issues. Moreover, the players are no longer restricted to one device and can play games anywhere and anytime.

Cloud gaming has already generated a great deal of interest among entrepreneurs, venture capitalists, and the general public. Currently several cloud gaming services with a variety of realizations are available on the market: OnLive [2], StreamMyGame [4], Gaikai [5], G-Cluster [6], OTOY [7], Spoon [8], and T5-Labs [9]. Some of these services are only accessible on a PC (either with specific application like OnLive or via browser-based applications like GaiKai), while others can be accessed via a TV with a set-top box. But a natural combination is that of Cloud gaming and mobile gaming; i.e., mobile clients playing games rendered as video and streamed to the mobile device. This will be particularly important in the future considering the importance of mobile gaming: 78.6 million people in the U.S. played mobile games in 2009, and downloads of mobile games increased tenfold compared to 2003 [10], while more than 125 million people played mobile games in the UK and US in 2012 [11]. While gaming in general is growing at a rapid pace, mobile gaming is seeing the fastest growth [12], and ABI Research reports that mobile gaming revenues will grow from US$5 billion in 2011 to more than US$16 billion in 2016 [13]. Indeed, the growth of mobile gaming is happening so quickly that industry observers believe mobile games will change the basis of competition in the gaming industry [10]. It is therefore obvious

NOSSDAV'13, February 26-March 1, 2013, Oslo, Norway.
Copyright 2013 ACM 978-1-4503-1892-1/13/02...$15.00.

that mobile games will continue to grow in terms of usage, popularity, and applications, and cloud gaming will likely become an important enabling technology to further spread the usage of mobile gaming.

While cloud gaming may reduce hardware costs for users and increase the revenue for developers by leaving out the retail chain, it also raises new challenges for service quality in terms of bandwidth and latency for the underlying network. In conventional online gaming, most if not all of the rendering is done at the client and therefore the network does not have any influence on the presentation, other than interaction with remote players. However this is not the case for GaV gaming where the streaming data rates are significantly higher than conventional gaming and similar to video streaming [14] and moreover, available network bandwidth greatly affects the player's quality of experience (QoE) [15]. This leads to a stringent and also game-dependent [16] requirement of network bandwidth for cloud gaming services and calls for appropriate methods of controlling the streaming bit rate.

In this paper, our objective is to introduce a content adaptation scheme for game scenes based on selective object encoding to reduce the *network bandwidth* and *processing power* requirements of real-time video encoding and streaming in cloud gaming systems. Since we are dealing with lots of simultaneous games and massive number of clients in the cloud, even a small amount of improvement in the encoding speed and transmission bandwidth can lead to significant savings overall. We therefore propose an approach that speeds up video encoding per game frame and also reduces the ensuing video’s bandwidth.

The rest of this paper is organized as follows: in section II, a brief review of the existing literature on video content adaptation is presented. Description of our proposed method of object selection and game scene content adaptation is in section III, followed by an experimental evaluation of this method and measurement of performance improvements in section IV, and finally section V concludes the paper.

2. RELATED WORK

Since in GaV the game is streamed as video to the client, it is important to understand video adaption techniques in mobile environments. During the past decade, a vast literature on video adaptation has been generated, which offer a rich body of techniques for answering challenging questions in pervasive media applications. 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 [16].

A general conceptual framework to model video entity, adaptation, resource, utility, and relations among them could be found in [18]. 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, [19] 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. Another branch of the research in the area of video content adaptation is focused on visual attention modeling [20] and attention-based adaptation schemes [21] which not only improves the perceptual quality but also saves the bandwidth and computation.

Reducing the complexity and hence speeding up the video encoding process is another area of research. Modern video coding standards like H.264/AVC [22] include many features to achieve much better rate-distortion efficiency and subjective quality, but the high computational complexity is the penalty. Such high requirement of computational resources leads to high power consumption. [23] presents a system level complexity reduction for H.264 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 a ROI classification method.

Despite the huge literature on video content adaptation, practical methods which are suitable specifically for cloud gaming are rare [24] and there are still many open issues in this research area. To the best of our knowledge, no other work has attempted to adapt the video of a game according to its context, as seen from a player’s perspective, in order to reduce energy and bandwidth consumption in the cloud. In the next section, we present our approach to achieve this goal.

3. PROPOSED METHOD

Our method is based on our previous successful experience with activity-based object selection and prioritization for 3D object streaming introduced in [25] and developed in [26]. However, unlike those 3D object streaming systems which require the client to render the scene using 3D objects received from the server, in the newly emerged cloud gaming systems, rendering and video encoding are to be done on the server side and only the encoded video will be streamed to the client. Therefore, in this work our goal changes to adapt the game scene in order to achieve a lower video bit rate and faster encoding time at the server side.

Algorithm 1: Gameplay Video Content Adaptation Algorithm

```
Input: $priorityIndex[]$
//An array containing normalized importance factors of all objects

$act \leftarrow getCurrentActivity()$;
//using activity recognizer component (virtual sensors)
$listOfAvailableSceneObjects \leftarrow getSceneObjects\ ()$;

$L \leftarrow length\ (listOfAvailableSceneObjects)$;
for $i \leftarrow 0$ to $L-1$ do
    $priorityIndex[i] \leftarrow normalized(listOfAvailableSceneObjects[i])$;
$listOfSelectedObjects \leftarrow$
$optimize(listOfAvailableObjects, priorityIndex, act)$;

$L \leftarrow length\ (listOfSelectedObjects)$;
for $i \leftarrow 1$ to $L-1$ do
    $videoFrame \leftarrow render\ (listOfSelectedObjects[i])$;
    //render and append the selected objects into the scene

$encodedVideoFrame \leftarrow encode(viedoFrame)$
//encode the raw video frame using x264

$stream\ (encodedVideoFrame)$;
//stream the encoded video frame to the client
```

The key idea in our approach is to exclude less important objects from the game scene so that it takes less processing time for the server to render and encode the frames, and furthermore a lower bit rate would be achieved to stream the resulting video. As described in [25] and [26], we propose an activity-based object selection algorithm which maintains a list containing the importance of each object, designed by game designers who know exactly what the game context is and what the importance of each object is for different activities inside the game. In each frame of the gameplay, importance of each object is evaluated based on the current activity of the player (activity can be walking, running, aiming, shooting, etc.). As described in [26], recognition of the player's action is carried out by utilizing a number of virtual sensors in the game. In fact, this task is implemented inside the *getCurrentActivity()* function in the above algorithm.

Using normalized importance factors, we prioritize the objects and, depending on the mobile device's capabilities, include a number of the most important objects in the current game scene. We have shown that removing objects that are irrelevant or less important to the current activity of the player does not significantly affect the user experience, including task accomplishment, fun, and immersion [26]. Having the prioritized list of objects in hand, we can then employ a number of selection and optimization criteria to choose the objects to be included in the scene and to be rendered. The algorithm is summarized in the listing of Algorithm 1. For more details about object importance, prioritized listing of objects, and selection of objects to be included in the game scene, we refer the readers to [25] and [26].

Using the prioritized listing of objects, in this work we propose to encode the video of the game scene with the more important objects and leave out the less important ones. In terms of encoding time, it is intuitive that this approach would reduce the processing time per frame: since there are fewer objects to add to the scene, there will be a reduction in the time needed to generate the overall image for the current frame.

In terms of video frame size, we also expect a reduction, due to the presence of fewer unique macroblocks in the video size, since each object likely has its own shape/color and adding more objects in the frame would require encoding more macroblocks that are visually different from other objects' macroblocks. This theory is tested in the next section where we show the amount of improvement in video encoding time and bit rate.

4. EVALUATION

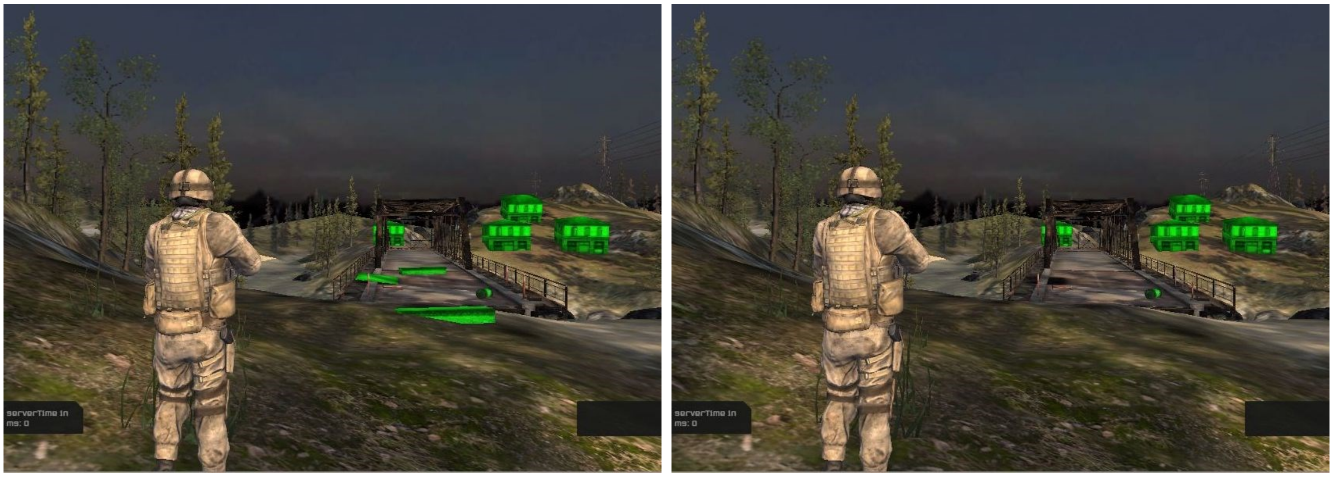
The proposed approach has been applied to two third-person shooter games, namely Bootcamp and AngryBots, which come with Unity 3D game engine [27].

In a scene of the Bootcamp game that the player needs to pass a bridge and shoot at an number of targets, we have implemented the object selection and scene customization method to purge the scene from unimportant objects. When the player is trying to aim at the targets, objects designed as walking obstacles are irrelevant to the player's activity and could be removed from the scene with minimum negative effect on player's experience. Figure 1 shows the scene of the Bootcamp game in the original and adapted versions. As can be seen, obstacles are missing in the adapted version of the game. We played the game starting from this scene for 30 seconds and captured the video output of the gameplay in raw AVI format. The captured video had 1064 pixels width and 948 pixels height and was recorded with 30 frames per seconds. Then, we encoded both recorded videos using the parameters listed in the Table 1 and two standard rate control methods, namely *CRF*: Constant Rate Factor and *ABR*: Average Bit Rate, of the x264 implementation of H.264/AVC video coding algorithm on a Dell Studio XPS 8100 and recorded the CPU time spent for this process using Intel VTune Amplifier [28].

We are interested in the size of the coded videos and their streaming bit rate to find out how much improvement could be made by removing the less important objects. Both average and peak streaming bit rates are employed as evaluation metrics. Note that if the throughput is bound by the users' connection, peak bit rate would be a critical factor in the quality of experience. Moreover, the CPU time taken by the server to encode the videos is another factor for potential improvement.

Table 1: Parameters of Video Encoding using x264

Name of the Parameter	x264 Function Argument	Value
Rate Control Method	rc.i_rc_method	X264_RC_ABR/CRF
Profile	Profile	High
Bit Rate	rc.i_bitrate	1 Mbps
Level of IDC	i_level_idc	0
Sub Pixel Refine Value	analyse.i_subpel_refine	2
Enable Motion Estimation for Chroma	analyse.b_chroma_me	0 (No)
Maximum Rate of VBV Buffer	rc.i_vbv_max_bitrate	2
Initial QP	rc.i_qp_constant	23
Maximum Interval Between IDR Frames	Keyint	0x00003FFF
Number of Frames per Second	i_fps_num	30
Enable CABAC (Context Adaptive Binary Arithmetic Coder)	b_cabac	0 (No)
Number of B frames	I_bframe	0
Enable PSNR	analyse.b_psnr	1 (Yes)
Enable SSIM	analyse.b_ssim	1 (Yes)



(a) Original Scene

(b) Adapted Scene

Figure 1: Bootcamp game: Obstacles (which are irrelevant for the "aiming" action) are present in the original scene (a) but missing in the adapted scene (b)

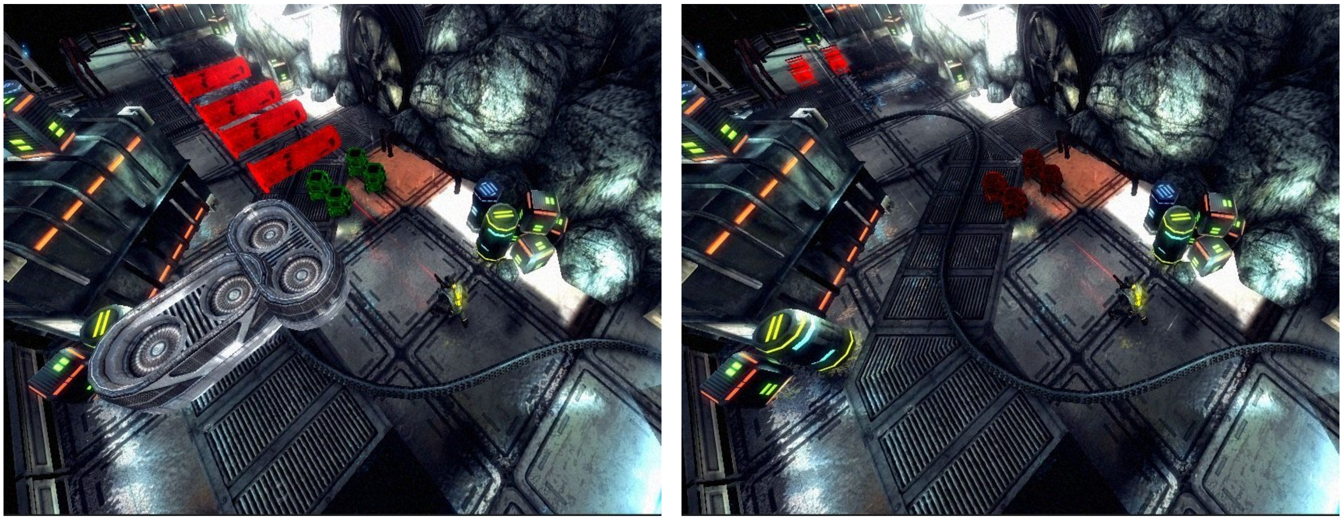
Table 2: Comparison of Streaming Bit Rate and Processing Time for the Bootcamp game

	CRF Rate Control Method			**ABR Rate Control Method**		
	Original Video	Adapted Video	Percentage Decrease	Original Video	Adapted Video	Percentage Decrease
Average Coded Frame Size	4051 B	3695 B	8.8%	4237 B	4079 B	3.7%
Average Streaming Bit Rate	949.4 Kbps	866.0 Kbps	8.8%	993.0 Kbps	956.0 Kbps	3.7%
Peak Streaming Bit Rate	15995 Kbps	15803 Kbps	1.2%	10776 Kbps	12071 Kbps	--
Average Encoding Time per Frame	51 ms	47 ms	7.8%	52 ms	50 ms	3.8%

Table 2 summarizes and compares these results for both scenarios. Having computed the size of each encoded frame for both videos, we calculated the average frame size of the coded videos. As can be seen in Table 2, for the CRF method, there is a considerable decrease of 8.8% in the average frame size after applying our adaptation method. The ABR method also results in 3.7% decrease in frame size. According to that, the average streaming bit rate of the video has decreased by a rate of 3.7-8.8% for different rate controls. While the savings are not huge for the video stream to a single client, we must consider that the cloud will be supporting a massive number of clients and, as we argued earlier, even a small saving per client will lead to an overall saving that is significant. This confirms the efficiency of our proposed method in terms of relaxing the network bandwidth requirements for real-time video streaming of the gamepaly. We can also see from the table that there seems to be no significant improvement or degradation of the peak streaming bit rate.

There is also a considerable improvement in encoding time from the viewpoint of the server's processors. Again, while the saving might seem negligible per frame, it leads to noticeable saving when we take into account 20 to 30 frames per second for a player, and then multiply that by the massive number of online gamers utilizing the cloud.

The second game we evaluated was AngryBots. Figure 2 shows the screenshots of the played scene. In the original scene, there are a number of obstacles which are not important when the player is going to aim and shoot at the targets. Using our method, when the activity recognition component of the game realizes that the current activity of the player is "aiming", these obstacles are considered as irrelevant objects and removed from the scene, as illustrated in the adapted version of the scene. Instead, some other targets, which are far from the player and are missing in the original scene (due to the conventional distance-based object streaming approach), are added to the adapted scene. This improves the player's quality of experience, especially in terms of task accomplishment.



(a) Original Scene

(b) Adapted Scene

Figure 2: AngryBots game: Obstacles (which are irrelevant for the "aiming" action) are present in the original scene (a) but missing in the adapted scene (b)

Table 3: Comparison of Streaming Bit Rate and Processing Time for the AngryBots game

	CRF Rate Control Method			**ABR Rate Control Method**		
	Original Video	Adapted Video	Percentage Decrease	Original Video	Adapted Video	Percentage Decrease
Average Coded Frame Size	4084 B	3993 B	2.2%	4107 B	3778 B	8.0%
Average Streaming Bit Rate	957.2 Kbps	935.8 Kbps	2.2%	962.6 Kbps	885.5 Kbps	8.0%
Peak Streaming Bit Rate	7469 Kbps	7299 Kbps	2.3%	4609 Kbps	4396 Kbps	4.6%
Average Encoding Time per Frame	62.5 ms	61 ms	2.4%	58 ms	54 ms	6.9%

This scene of the game was played for 20 seconds with the same sequence of player's actions. Again the video output of both versions of the game were captured in raw AVI format with a size of 1360x636 pixels and 30 frames per second.

Table 3 shows the results of the H.264/AVC video encoding process of the recorded videos using CRF and ABR rate control methods on a Dell Studio XPS 8100. Again we see that the proposed content adaptation method results in a considerable improvement (2.2-8.0%) in the streaming bit rate as well as the required encoding time per frame (2.4-6.9%).

Although we measured the above performance metrics using our specific approach for object prioritization and selection, it should be noted that our results could be generalized. In other words, we showed that in a "Game as Video" scenario, removing some of the game objects (using any approach, not just our approach) will indeed lead to significantly lower bit rate for the ensuing video. This result by itself is an important contribution and could encourage other researchers to come up with other techniques to reduce the number of objects to be rendered in the cloud. In addition, we showed that such an approach would also lead to faster encoding time, although the improvement in encoding time is less significant than that of bit rate.

5. CONCLUSION

We presented a content adaptation scheme for game scene adaptation using an object selection and optimization method, such that only the most important objects from the perspective of the player's activity are encoded in the scene and irrelevant or less important objects are omitted. Experimental results showed that this approach would achieve a significantly lower streaming bit rate (between 2.2% to 8.8% less than the original video) and slightly less processing time on server side. It should be mentioned that our method is complementary to existing methods, such as low-polygonal modeling and level of detail scaling. We do not intend to replace those methods, but present a new method that can lead to even more bandwidth and energy saving at the sever side.

A subjective evaluation of the quality of experience is planned to be carried out in future work to verify whether the players believe the QoE is maintained, although our previous work has shown that to be the case for client-side rendering [26], and so we do not expect a different result for server-side rendering since the final output of both approaches looks the same to the player. Moreover, a comparison of QoE between the proposed scheme and the strategy of higher compression of the entire scene with all objects (using scalable codecs for instance) could be another interesting research area.

It should be noted that there are some other alternatives for how to deal with the less important objects in a scene of the game. In the present work we completely removed such objects; however, we could have rendered them with a lower level of details (LoD) and/or encoded regions containing those objects with lower bit rates. For future works, we intend to examine the above-mentioned alternatives instead of completely excluding objects from the scene. After performing additional experiments, we would be able to provide a detailed analysis and comparison of the impact of different levels of adaptation on bit rate reduction for GaV.

Another research direction for the future work is to come up with energy-aware video encoding algorithms which are capable of taking into account the constraints imposed by bearable complexity of either or both encoder (server) and decoder (client). Note that the complexity of server-side encoding of a massive number of game sessions could be restricted by the Cloud's available processing facilities and the manageable complexity of decoding is generally limited by the processor of the thin clients.

6. REFERENCES

[1] Kuan-Ta Chen, Yu-Chun Chang, Po-Han Tseng, Chun-Ying Huang, and Chin-Laung Lei, “Measuring the latency of cloud gaming systems,” in Proc. 19th ACM Intl. Conference on Multimedia (MM '11), ACM, New York, NY, USA, pp.1269-1272, 2011.

[2] OnLive. [Online]. http://www.onlive.com/

[3] CiiNOW. [Online]. http://www.ciinow.com/

[4] StreamMyGame. [Online]. http://www.streammygame.com/

[5] GaiKai. [Online]. http://www.gaikai.com/

[6] G-Cluster. [Online]. http://www.gcluster.com/

[7] OTOY. [Online]. http://www.otoy.com/

[8] Spoon. [Online]. http://www.spoon.net/

[9] T5-Labs. [Online]. http://www.t5labs.com/

[10] J. Soh, B. Tan, “Mobile Gaming,” *Communications of the ACM*, 51(3): 35 - 39, March 2008.

[11] Information Solutions Group, “PopCap Games Mobile Gaming Research - 2012”, June 15 2012, [Online]. http://www.infosolutionsgroup.com/popcapmobile2012.pdf/

[12] N. Bilton, “Video Game Industry Continues Major Growth, Gartner Says,” *The New York Times*, July 5 2011, [Online]. http://bits.blogs.nytimes.com/2011/07/05/video-game-industry-continues-major-growth-gartner-says/

[13] ABI Research Report, “Mobile Gaming”, July 27 2011, [Online]. http://www.abiresearch.com/research/1006314/

[14] M. Claypool, D. Finkel, A. Grant, and M. Solano, “Thin to Win? Network Performance Analysis of the OnLive Thin Client Game System,” in Proc. of the 11th ACM/IEEE NetGames, Venice, Italy, Nov. 2012.

[15] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoβfeld, “An Evaluation of QoE in Cloud Gaming Based on Subjective Tests,” in Proc. 5th Intl. Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE Computer Society, Washington, DC, USA, pp. 330-335, 2011.

[16] Yeng-Ting Lee, Kuan-Ta Chen, Han-I Su, and Chin-Laung Lei, “Are All Games Equally Cloud-Gaming-Friendly? An Electromyographic Approach,” in Proc. of the 11th ACM/ IEEE NetGames, Venice, Italy, Nov. 2012.

[17] S.-F. Chang, A. Vetro, “Video Adaptation: Concepts, Technologies, and Open Issues,” *Proceedings of the IEEE* , vol.93, no.1, pp.148-158, Jan. 2005.

[18] Jae-Gon Kim, Yong Wang, Shih-Fu Chang, “Content-adaptive utility-based video adaptation,” in Proc. 2003 Intl. Conference on Multimedia and Expo (ICME'03), vol.3, pp. 281-284, 6-9 July 2003.

[19] W.-H. Cheng, C.-W. Wang, J.-L. Wu, “Video Adaptation for Small Display Based on Content Recomposition,” *Circuits and Systems for Video Technology, IEEE Transactions on* , vol.17, no.1, pp.43-58, Jan. 2007.

[20] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, Hong-Jiang Zhang, He-Qin Zhou, “A visual attention model for adapting images on small displays,” *Multimedia Systems*, vol. 9, pp. 353-364, 2003.

[21] Y. Wang, H. Li, X. Fan, 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, no. 4, pp. 565-584, 2006.

[22] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst.Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[23] A.K. Kannur and B. Li, “Power-aware content-adaptive H.264 video encoding,” in Proc. IEEE Intl. Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 925-928, 2009.

[24] Shu Shi, Cheng-Hsin Hsu, K. Nahrstedt, and R. Campbell, “Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming,” in Proc. 19th ACM Intl. Conference on Multimedia, ACM, New York, NY, USA, pp.103-112, 2011.

[25] H. Rahimi, A. A. Nazari S., and S. Shirmohammadi, “Context-Aware Prioritized Game Streaming,” in Proc. of Workshop on Interactive Ambient Intelligence Multimedia Environments, in Proc. IEEE International Conference on Multimedia & Expo (ICME 2011), Barcelona, Spain, July 11-15, 2011.

[26] H. Rahimi, A. A. Nazari S., and S. Shirmohammadi, “Activity-Centric Streaming of Virtual Environments and Games to Mobile Devices”, in Proc. IEEE Symposium on Haptic Audio Visual Environments and Games (HAVE 2011), Qinhuangdao, Hebei, China, October 15-16, 2011, pp. 45 – 50.

[27] Unity 3D Game Engine. [Online]. http://www.unity3d.com/

[28] Intel VTune Amplifier. [Online]. http://software.intel.com/en-us/articles/intel-vtune-amplifier-xe/