2012 IEEE 4th International Conference on Cloud Computing Technology and Science

Multiplayer Cloud Gaming System with Cooperative Video Sharing

Wei Cai and Victor C.M. Leung
Department of Electrical and Computer Engineering
The University of British Columbia
Vancouver, Canada V6T 1Z4
Email: {weicai, vleung}@ece.ubc.ca

Abstract—**Mobile cloud-based video gaming (MCVG) is an emerging trend in moving the online entertainment industry into the cloud era. In MCVG, the game engines are hosted in the cloud, and the rendered gaming videos are transmitted over wireless networks to the mobile devices. In reverse, game players' interactions on screens are sent to the cloud server over the same networks. How to compress and transmit the real-time gaming video, so that during the gaming session, the expected server transmission rate over the bandwidth-limited wireless network is minimized while satisfying the quality of experience demanded by the players, is a great technical challenge that is addressed in this paper in a multi-player gaming context. We exploit the correlations between the gaming videos for distinct players in the same gaming scene to propose a cloud gaming system with cooperative video sharing, in which the cloud game server is able to efficiently encode and transmit multiple video streams to a group of players, while those players are able to decode their video in a cooperative manner by sharing contents via a secondary network such as ad hoc wireless local area network. Experimental results show that the expected server transmission rate can be significantly reduced compared to the conventional video encoding schemes for cloud games.**

Keywords-**cloud; game; video; network; encoding; cooperative**

I. Introduction

Mobile games contribute a huge number of downloads and consequently large potential profits in the application market. However, the constraints of hardware in mobile devices, such as central processing unit (CPU), graphic processing unit (GPU) and storage, limit the representation of games. Advanced graphical technologies, such as three dimensional (3D) scene rendering, are intensive CPU consumer and battery drainer [1]. Consequently, mobile cloud computing [2] is attracting much attention as a promising technology to enable video games as a service [3]. Since the games are rendered in cloud servers, a less capable computer or mobile device may be used to support any kind of game as long as it is able to play the video.

Industry is now leading the way in this area. OnLive[1], Gaikai[2] and G-Cluster[3] are commercial providers of cloud-based video games. They started the business from desktop solutions, in which the game engines are hosted in the cloud and the game videos are distributed to the desktop screens for the players via the Internet. Recently, OnLive has expanded their market to mobile devices, such as Android and iOS platforms, while Gaikai provides platform-independent game service on browser-based technologies.

However, those cloud-based video games still suffered from the bandwidth-bottleneck of Internet access. The bandwidth constraints restrict the bit rate of gaming videos, while the jitter and delay affect the quality of experience (QoE) for the players. Therefore, efficiently encoding and transmitting real-time gaming videos become the most critical issues in cloud-based video gaming system. There has been plenty of work focusing on scalable video encoding for cloud games, e.g., [4]. However, these solutions intrinsically sacrifice the video quality to guarantee the player QoE.

Another trend for the game industry is online multiplayer gaming. Game players are no longer satisfied with enjoying the games alone but they prefer to connect to others. The interactions between players bring more challenges in the design of game servers. However, it also motivates us to investigate the potential benefits of video sharing among collocated players involved in the same scene.

In this work, we explore the idea of peer-to-peer sharing between multiple players in the same game scene and propose a multiplayer cloud gaming system with cooperative video sharing, in which the mobile devices are connected to the cloud server for real-time interactive game videos, while sharing the received video frames with their peers locally via ad hoc network connections. To the best our knowledge, our work is the first in cloud-based mobile video gaming that exploit peer-to-peer cooperative video sharing to substantially reduce the transmission rate from the cloud server to the game clients, thus overcoming the bandwidth-bottleneck of Internet access.

The reminder of this paper is organized as follows. Section II summarizes the related work. Section III studies the correlations of gaming videos between distinct players in the same game. Section IV proposes and models the cooperative video sharing system for the cloud games, in which the cloud game server is able to efficiently encode and transmit multiple video streams to a group of players, while those players are able to decode their video in cooperative pattern via a secondary network such as ad hoc wireless

[1] http://www.onlive.com
[2] http://www.gaikai.com
[3] http://www.gcluster.com

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

local area network (WLAN) for content sharing. In Section V, we design a video encoder in the cloud, which can efficiently encode the multiple video streams for distinct players. Simulation results are presented and analyses in Section VI. Lastly, Section VII concludes our work.

II. Related Work

The original motivation for our work is to reduce the server transmission rate by sharing received video frames. Therefore, the correlations of video frames have critical impacts on the system's performance. This topic has been studied for streaming in light field [5] and multiview [6] video.

Light field is a large set of spatially correlated images of the same static scene captured using a two dimensional (2D) array of closely spaced cameras. The correlations of light field images are studied in [7], which indicates the correlation between two different views to a static scene is related to the geographical distance between them.

More similar to the cloud-based video gaming case, interactive multiview video switching [8] designs a pre-encoded frame representation of a multiview sequence for a streaming server, so that streaming clients can periodically request desired views for successive video frames in time. P-frames and Distributed Source Coding (DSC) frames are inter frames used to explore correlations between neighbouring views while users switch their views from one to another.

However, compared to light field and multiview switching, the modelling of correlations between players' views is more complicated. There is an infinite number of views as the players adjust their personal views while they are walking through the scene and participating a battle field. The dynamic switching makes the correlation hard to predict.

Another significant difference to the above light field and multiview switching work is that, encoding the video for cloud games is essentially a real-time encoding process. The cloud renders the game representations into video frames and transmits them to the mobile clients. The encoding starts with an intra-frame (I-frame), followed by a certain number of inter frames, such as P-frame [9], B-frame, DSC frames [10], and repeats. Therefore, how to determine the sequence of various types of frames become the most critical problems in video encoding, in order to achieve the tradeoff between bit rate and error rate. In recent video encoding research, the GOP (Group of Pictures) [11] length is set to be adaptive, which implies a structure with one I-frame and variable number of inter frames.

III. Correlation of Video Frames

In this section, we study the frame size for two types of P-frames: intra-video P-frame and inter-video P-frame.

Fig. 1 shows an example of frame dependency in a four player gaming scenario. As depicted, intra-video P-frames are those predicted from previous frames in the video of a specific players game, while inter-video P-frames are those predicted from video frames of peers games.

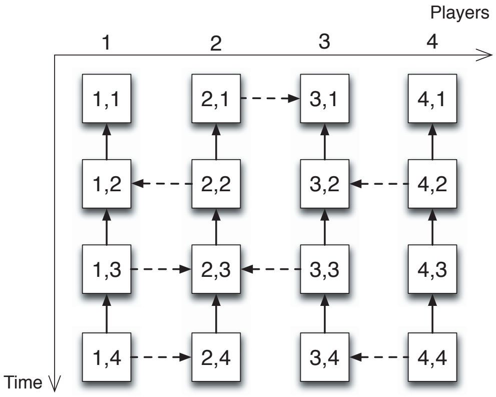

Figure 1. Frames correlation in real-time multiplayer game videos

A. Intra-video P-frame

The frame size of intra-video P-frame is subjected to the variance ratio of the game video content. Apparently, if the avatar is experiencing a more dynamic scene, the resulting P-frame size will be larger. For example, when the players are participating in a large Diablo battle field, where many wizards and demon hunters are throwing magnificent full-screen magic, the change of video content will be dramatic. In this work, we assume the size of intra-video P-frames P_{intra} follows a Poisson distribution:

$$f(P_{intra}) = \frac{\mu^{P_{intra}} e^{-\mu}}{P_{intra}!} \tag{1}$$

where the mean frame size is $E(P_{intra})$ is μ.

B. Inter-video P-frame

The frame size of inter-video P-frame, P_{inter}, is subjected to the correlation between two videos for two game players. Before making assumption for P_{inter}, we have to discuss the types of multiplayer games: first-person game, second-person game and third-person game. In video games, first-person refers to a graphical perspective rendered from the viewpoint of the player character, e.g., Counter-Strike. The second-person is similar to first-person but rendered from the back of the player character, which means the player can see their avatar on the screen, e.g., Grand Theft Auto. In contrast, third-person games always have a fixed angle of view, so-called God-view; the players watch the whole game scene in a bird-view, so that they can easily catch the surrounding environment of their avatars. Classic third-person games include Diablo, Command & Conquer, FreeStyle, etc.

In contrast to the cases of first-person and second-person, the correlation model for third-person games is much simpler: the players are viewing very similar videos when their avatars are geographically close to each other. To

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

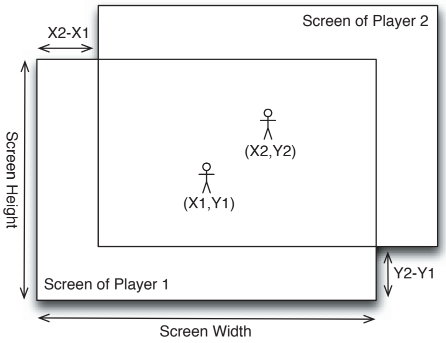

Figure 2. Correlation of inter-video frames

simplify the system, we consider third-person games in this work. Since the players are all in God-view, we can simply calculate the overlap of the two videos according to the positions of the avatars, given most of the third-person games rolls the map with an avatar-centric mechanism. Fig. 2 demonstrates two correlated videos for player 1 and player 2. Hence, we can derive the overlap ratio $R_{overlap}$ as:

$$R_{overlap} = \frac{[W - (|X_2 - X_1|)][H - (|Y_2 - Y_1|)]}{WH} \quad (2)$$

where H is the screen height, W is the screen width, (X_1, Y_1) and (X_2, Y_2) denote the coordinates of the two players in the gaming map.

Based on the analysis, we formulate the size of inter-video P-frames P_{inter} as follows:

$$P_{inter} = (1 - R_{overlap})I\rho \quad (3)$$

where I is the size of an I-frame and ρ is the compression ratio the encoder is able to achieve.

IV. System Modelling

A. System Overview

The system model we propose for the multiplayer cloud gaming system with cooperative video sharing is shown in Fig. 3. Game engines are hosted in the cloud to handle the commands from players. To facilitate multiplayer interactions, these game engines are connected to a Multiplayer Game Server in conventional fashion.

In contrary to existing cloud gaming systems, the proposed system adds a Video Encoder Server as a gateway for the game videos. The Video Encoder Server is able to explore the correlations between video frames and encode the video streams for multiple players in a centralized model, minimizing server transmission rate. Note that the mobile devices are connecting to each other locally via a secondary network, e.g., an ad hoc WLAN, in order to share the video frames they have received from the cloud server. The bandwidth of the ad hoc WLAN is assumed to be sufficiently large for all mobile devices in the immediate neighbourhood to share their video frames when needed. Thus, the bandwidth constraint in the ad hoc WLAN is not explicitly modeled.

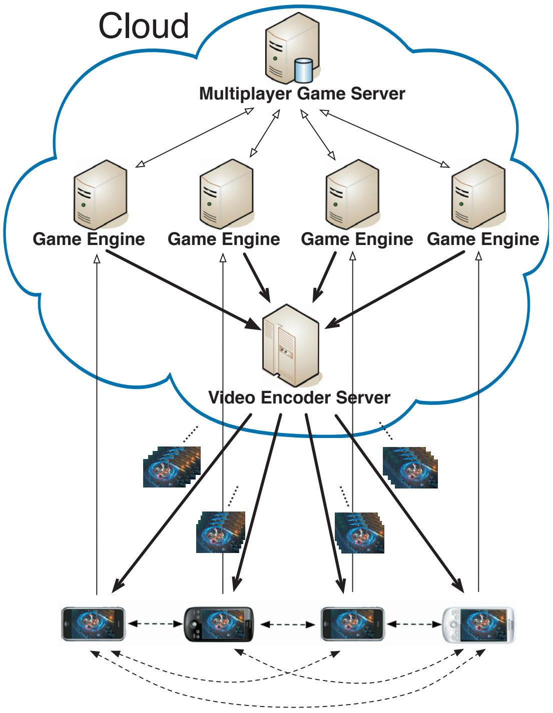

Figure 3. Overview: Cooperative cloud-based video gaming system

B. Gaming Map Model

As discussed in Section III, our work focusses on the third-person game. From the God-view, all kind of complicated 3D scenes are represented as 2D illustrations. Hence, we simply define the gaming map as a 2D space of M screen size $\times M$ screen size.

We assume the player's moves in the map are in steps of K pixels. Let W and H denote the width and height of the gaming screen. Therefore, the players are able to move their avatars in coordinate (X, Y), where $X \in [0, \frac{MW}{K}], Y \in [0, \frac{MH}{K}]$. However, if we restrict the avatar to the centre of the screen, the moving area will be restricted to (X, Y), where $X \in [0, \frac{(M-1)W}{K}], Y \in [0, \frac{(M-1)H}{K}]$. In this work, we adopt the later model.

C. Player Interaction Model

Based on the map model, we model the N players' interactions in the same game scene. Specifically, the avatars in the game scenes are able to perform two kinds of movements: random walk and group chase. This is from an observation of multiplayer games: players are freely hunting in the world; however, they are also prone to gather together with their team mates or opponents, in order to perform teamwork or competition.

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

For random walk movements, each avatar is moving towards a distinct and contiguous trajectory in the map. The probability for the avatar to conduct random walk mode is denoted as p_{rw}. In this work, we assume the players are able to stay still or move their characters to one of N_{adj} adjacent directions with equal probability. Therefore, the probability to each adjacent view is $\frac{p_{rw}}{N_{adj}+1}$.

For group chase movements, the avatars randomly select another avatar in the scene and approach it over a certain period of time T_{chase}. Let the probability of a group chase movement be $p_{gc} = 1 - p_{rw}$, and the probability of any one the $N - 1$ avatars to be selected is equal and give by $p_{appr} = \frac{1}{N-1}$.

V. Design of Video Encoder

In this section, we design the featured video encoder server, which efficiently encodes and distributes the videos for multiple players in real-time. Since the encoding is a centralized approach within the cloud, it is easy to gather information from all of the players and derive the optimal encoding solution to minimize the server transmission rate.

A. Optimal Encoding Algorithm

In order to achieve optimal encoding, we adopt a greedy approach such that for each video frame for a distinct game player, the cloud-based encoder is able to compare and select the smallest frame from the three types of encoding solutions, including I-frames, Intra-video P-frame and Inter-video P-frame. The selection procedure can be achieved in several steps:

1) Frame Size Estimation: The first step estimates the frame sizes for intra-video and inter-video P-frames. For N game players, the encoder estimates frame sizes $P_{intra}[N]$ for N intra-video P-frames, where $P_{intra}[i]$ represents the frame size of intra-video P-frames for ith player. Meanwhile, we store the estimated frame-size of inter-video P-frames for all pairs of players in a matrix $P_{inter}[N][N]$, where $P_{inter}[i][j]$ represents the frame size of inter-video P-frame for ith player which predicts to the video frame of jth player.

2) Grouping: Based on the analysis in Section III, the N players might form several groups, given a sufficiently large game map. Hence, we need to separate these videos into groups, so that the encoder is able to optimize the video output for each group. We eliminate those $P_{inter}[i][j]$ which are larger than $P_{intra}[i]$, thus, the frames kept in P_{inter} are "efficiently correlated". Afterwards, we enumerate the array of P_{inter} to group mth and nth players into same group if $P_{inter}[n][m]$ exists.

3) Optimal Encoding: In each efficiently correlated group, the encoder should able to perform optimal encoding to minimize the expected server transmission rate. One and only one video frame in a group should be encoded as an intra-video P-frame, while others are encoded as inter-video P-frames that minimize the transmission rate. We iteratively select one video frame to be encoded as intra-video P-frame and calculate an inter-video P-frame structure with minimum server transmission. At last, we choose the best solution among all results. In each iteration, the encoder encodes a different video frames $F[k]$ as intra-video P-frame and creates a directed, weighted graph with the set of P_{inter}. Therefore, finding an optimal solution with minimum transmission cost is to find a minimum spanning tree with the root of the $F[k]$. In this work, we implement the procedure with Prim algorithm.

B. Multi-hop Decoding Problem

With the proposed encoder, we are able to find the optimal solution for inter-video encoding. However, there is a practical problem that needs to be addressed when we consider a realistic system: the encoder groups efficiently correlated video frames and construct a tree to describe the dependency of video frames, there might be a need multi-hop decoding when the encoded video frames are sent to the mobile devices. For example, when a frame for player 1 is encoded as $P_{intra}[1]$, and the video frames for player 2 and 3 are encoded as inter-video frame $P_{inter}[2][1]$ and $P_{inter}[3][2]$, then the player 3 need to wait for player 2 to decode its frame by receiving the $P_{intra}[1]$ from player 1, so that the larger latency can be expected. However, gaming applications are very sensitive to latency; therefore, multi-hop decoding might affect the QoE for the players.

C. Solution to the Multi-hop Decoding Problem

To solve the multi-hop decoding problem, we propose another one-hop encoding algorithm in this section. The idea is basically a greedy approach: the encoder always searches for the video frames $F[x]$ which has the most dependent frames $P_{inter}[y][x]$ in the particular group, then encodes the $F[x]$ as $P_{intra}[x]$ and all $F[y]$ as $P_{inter}[y][x]$. The process continues until all video frames are encoded. With One-Hop Encoding Algorithm, we restrict the decoding into one-hop, which eliminates the multi-hop decoding problem as stated in the previous section. Thus, acceptable decoding latency is achieved.

VI. Performance Evaluations by Simulations

A. Experimental Setup

To evaluate the performance of our proposed system and encoding schemes, we set up the following experiments. For the video data, we downloaded the images from Stanford bunny light field set [12], each image of size 1024×1024. To encode I- and P-frames, we used the H.236-based codec in [10]. Quantization parameters were set so that the Peak Signal-to-Noise Ratio (PSNR) of the encoded frames was around 32dB. As described in Section III, we assume the P-frames for intra-video and inter-video follow designated distribution and formulation.

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

Default values for the parameters of the simulation were set as follows: number of players in the system N is 8, average game time T is 1000 second, fps (frames per second) is 24, and the game map is M screen size $\times M$ screen size, where M is 4; mean intra-video P-frame size μ is 28749 bytes, intra-encoded video frame size I is 245296 bytes, compression ratio for inter-video frame ρ is 0.7, screen width W is 1024 pixels, height H is 1024 pixels, avatar's each move K is 32 pixels, probability of random walk P_{rw} is 0.7, and time for each chase approach T_{chase} is 5 seconds; the avatar is able to move to 8 adjacent directions, which results in $N_{adj} = 8$.

We use the server transmission rate as the performance metric, since the proposed system aims to reduce the bandwidth requirement of the network access. We compare the performance of three schemes: original intra-video encoding, optimal inter-video encoding and one-hop restricted inter-video encoding. The original intra-video encoding is the simplified version of traditional video encoding, in which the system is able to encode the video frames with one I-frame and infinite successive P-frames ($N_{GOP} = \infty$), which achieves the lowest server-to-client bandwidth in gaming video transmission. Optimal inter-video encoding and one-hop restricted inter-video encoding are the schemes proposed in Section V.

B. Experimental Results

We perform a series of simulations to study the impacts of the parameters on the system and the proposed encoding schemes.

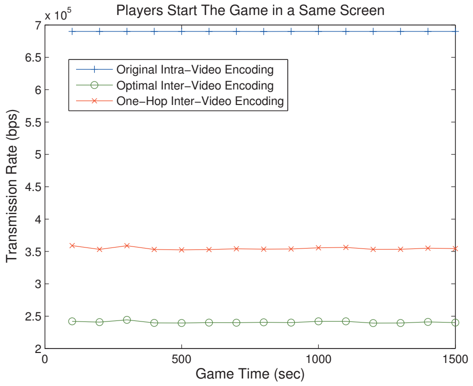

Figure 4. Players start the game in the same screen

The initial positions of the avatars are very sensitive elements for the system, since they are the key factors to determine the correlations between gaming videos. Fig. 4 illustrates the simulation results when all of the players are starting their games in the same screen. It is apparent that the optimal encoding scheme achieves the lowest server transmission rate, while the more practical one-hop restricted encoding scheme also yields a significant decrease in bandwidth as well.

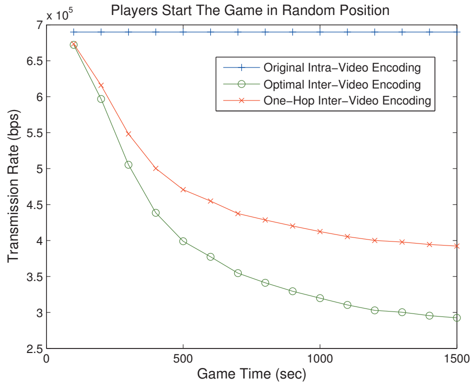

Figure 5. Players Start the Game in Random Position

In contrast, Fig. 5 shows the result for the case that each avatar chooses a random starting coordinate in the gaming map. The gain of video sharing is not significant at the beginning, since the avatars are randomly distributed in the map. However, the players are gathering together as the game progresses, which brings a significant reduction in the bandwidth requirement of the cloud server.

To better investigate the impact of players' interactions, we adopt the random starting model for the following experiments.

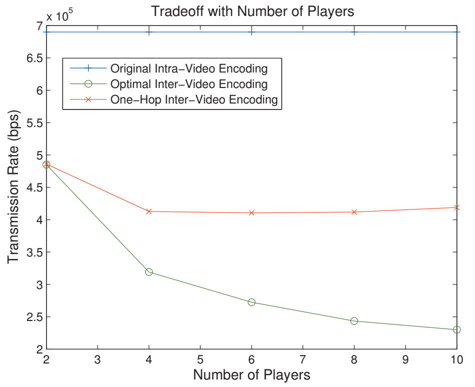

Figure 6. Tradeoff with number of players

Fig, 6 depicts the relations between transmission rate and the number of players in the system. With a fixed gaming map, a larger number of players increases the chance of correlated videos generated in the cloud server, therefore resulting in a lower transmission rate for each player. An interesting phenomenon from this figure is that, as the number of players increases, the decrease in transmission rate of the optimal inter-video encoding scheme is more significant than that of the one-hop restricted solution. The reason is that, multiple hop correlations between gaming

978-1-4673-4510-1/12/$31.00 ©2012 IEEE

videos are prone to occur when more players are connecting to the same game scene, which benefits the optimal scheme but not the one-hop restricted encoder.

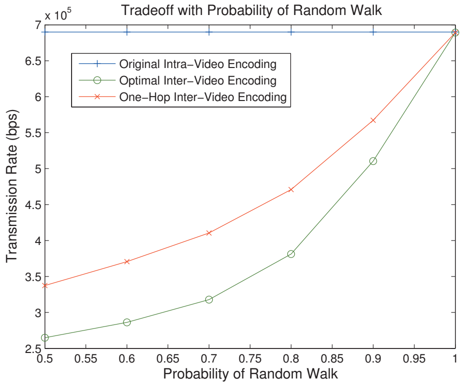

Figure 7. Tradeoff with probability of random walk

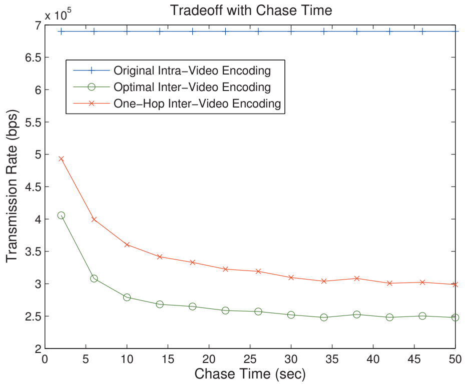

Figure 8. Tradeoff with time for each chase approach

Fig. 7 and Fig. 8 show the impacts of the behaviours of the players. A higher probability of random walk reduces the chance for the avatars to share similar videos. Therefore, the performances of the proposed encoding schemes are worse. Similarly, the longer time for each chase approach is, the easier for the players to share correlated video frames, thus resulting in a significant gain for the cooperative system.

If we set the probability of random walk to 1, which implies all moves for the avatars are totally random, then there is little correlation between different players videos to be exploited, which leads to an all intra-video encoding solution. Therefore, the gain of the proposed system is eliminated.

VII. Conclusion

In this paper, we have investigated the correlations of video frames for multiplayers in a cloud-based gaming system. Based on the analysis, a cooperative system is proposed for cloud game participants to share their received video frames, thus reducing the bandwidth required between the cloud server and the players. We have presented an optimal encoding solution for the cloud server and provided a one-hop restricted encoding scheme that is suitable for practical implementation. Extensive simulation experiments on multiplayer ARPG have been performed. Results show that the server transmission rate is reduced by up to 64% in the ideal case with the optimal encoding scheme, and up to 54% in the more practical case employing our one-hop restricted encoding solution.

Acknowledgement

This work is supported by a University of British Columbia Four Year Doctoral Fellowship and by funding from the Natural Sciences and Engineering Research Council.

References

[1] K. Yang, S. Ou, and H. Chen, “On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications,” *Communications Magazine, IEEE*, vol. 46, no. 1, pp. 56 –63, january 2008.

[2] W. Song and X. Su, “Review of mobile cloud computing,” in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, may 2011, pp. 1 –4.

[3] S. Wang and S. Dey, “Modeling and characterizing user experience in a cloud server based mobile gaming approach,” in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 30 2009-dec. 4 2009, pp. 1 –7.

[4] S. Wang and S. Dey, “Rendering adaptation to address communication and computation constraints in cloud mobile gaming,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, dec. 2010, pp. 1 –6.

[5] M. Levoy and P. Hanrahan, “Light field rendering,” in *ACM SIGGRAPH*, New Orleans, LA, August 1996.

[6] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, “Efficient prediction structures for multiview video coding,” in *IEEE Transactions on Circuits and Systems for Video Technology*, November 2007, vol. 17, no.11, pp. 1461–1473.

[7] W. Cai, G. Cheung, T. Kwon, and S.-J. Lee, “Optimized frame structure for interactive light field streaming with cooperative cache,” in *IEEE International Conf. on Multimedia and Expo*, Barcelona, Spain, July 2011.

[8] G. Cheung, N.-M. Cheung, and A. Ortega, “Optimized frame structure using distributed source coding for interactive multiview streaming,” in *IEEE International Conference on Image Processing*, Cairo, Egypt, November 2009.

[9] G. Cheung, A. Ortega, and N.-M. Cheung, “Generation of redundant coding structure for interactive multiview streaming,” in *Seventeenth International Packet Video Workshop*, Seattle, WA, May 2009.

[10] N.-M. Cheung, A. Ortega, and G. Cheung, “Distributed source coding techniques for interactive multiview video streaming,” in *27th Picture Coding Symposium*, Chicago, IL, May 2009.

[11] J. Lee, I. Shin, and H. Park, “Adaptive intra-frame assignment and bit-rate estimation for variable gop length in h.264,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 10, pp. 1271 –1279, oct. 2006.

[12] “Stanford Light Field,” http://lightfield.stanford.edu/lfs.html.

978-1-4673-4510-1/12/$31.00 ©2012 IEEE