

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/346716243>

EFFICIENT VIDEO ENCODING ACCELERATION FOR CLOUD GAMING

Article in Journal of Theoretical and Applied Information Technology · September 2020

CITATIONS

0

READS

89

2 authors:



Ahmad A. Mazhar

Higher Colleges of Technology

14 PUBLICATIONS 39 CITATIONS

SEE PROFILE



Manar Mizher

Amman Arab University

13 PUBLICATIONS 16 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Gaming techniques [View project](#)



Passive object-based video forgery detection [View project](#)

EFFICIENT VIDEO ENCODING ACCELERATION FOR CLOUD GAMING

AHMAD A. MAZHAR¹, MANAR A. MIZHER²

¹Algonquin College, Ottawa, Canada

²Amman Arab University, Amman, Jordan

E-mail: ¹ah.mazhar@yahoo.com, ²mmizher@aau.edu.jo

ABSTRACT

Cloud computing is an information technology model that provides access to system resources with higher level of services capability. These resources are considered reliable, flexible and affordable for many types of applications and users. Gaming industry is one field that gained benefits of cloud computing as new cloud gaming architectures have been introduced. Many advantages of cloud gaming have affected the success of gaming according to the improvements on traditional online gaming. However, cloud gaming suffers from several drawbacks such as the huge amount of required video processing and the computational complexity needed. This paper shows the original system drawbacks and devises a new and novel algorithm for speeding up the encoding process and reducing the computational complexity. Improvements on the video codec led to 41% speeding up on the total encoding time with negligible loss of users' satisfactions.

Keywords: *Cloud Gaming, Computational Complexity, Motion Estimation, H.264, Video Encoding*

1. INTRODUCTION

Video gaming is becoming more popular worldwide as the entertainment spreads widely in all societies. The Entertainment Software Association (ESA) reported that about 60% of Americans play video games regularly [1]. As a result, the revenue of gaming industry has grown-up to reach billions in last few years [2]. This growth is not the target as most of the game developers are improving their products to attract more game players. One of the most attractive opportunity for the developers is by exploiting cloud-computing techniques and emerging it with game development. The main idea is to deploy and fasten games using clouds that results so-called cloud gaming. Using clouds to process games instead of consoles or personal computers would end with faster games and wider reach of customers. Cloud gaming reduces the required amount of complex computational tasks that need to be executed on the client side. The huge process of rendering high definition video sequences would be conducted on the cloud and dealt with a game as video (GaV) technique [3].

Cloud gaming term has spread and become one of the most popular terms in on-line gaming. The

benefits of implementing cloud computing have increased the reliability of relying on cloud to conduct games. Basically, cloud gaming runs the game as real time application on a remote center and streams the desired actions to the user's device [4, 5] as this device is normally limited. In this technique, the user's device is only required to process limited amount of data that related to user actions and controlling but not required to process or rendering any video sequences. All videos are streamed to the client as video sequences and no need to render it. User needs only a video decoder to show the received game sequences. Users also receive the communication messages and instructions from the game server. This trend of cloud gaming was devised to minimize the dependency on the user's hardware. The main requirement to a successful enjoyable cloud game is the high-speed internet connection. Providentially, high speed internet is now available in most places around the world [6]. Although this technique reduces the processing load on the user side, a huge amount of data is to be communicated and a lot of bandwidth would be consumed.

The cloud gaming technique has many benefits to game developers, game service providers as well as gamers. The developers will guarantee that no piracy is threatening their games as the game is not downloaded. The service providers will gain more demand on their services and devise a new business model. Finally, gamers have many benefits such as: gaining access to the games anytime, saving money as they will not be required to upgrade their hardware in short periods and conducting tournaments and game sharing are easier and cheaper [7]. The CloudUnion is a successful example of cloud gaming platform. After users downloaded the client, the CloudUnion streams the game as video in different resolutions. The minimum bandwidth required is 2Mbps which is available in most countries worldwide. However, 6 Mbps will guarantee high quality game streaming [8].

As cloud gaming term has a unified idea of exploiting the cloud to improve gamers enjoyment. However, structures of exploiting this cloud is implemented in different ways. Three main categories can be illustrated in this field [9]. The first category depends on rendering all the computational operations at the server side. These operations contain gaming logic and video (rendering, capturing, encoding and streaming). This category is known as Remote Rendering GaaS (RR-GaaS). The client is not required to process any part of the game. It is only required to receive the video of the game, decode it, and display it to the user. The lower computational process on the client side is a major advantage of this technique. However, client may suffer from the end-to-end delay.

The second category relies more on the client side. Local Rendering GaaS (LR-GaaS) is the technology where the terminal is responsible for the whole processing operations. These operations include video coding/decoding and streaming. The limitations of this approach can be the high demand on the client capability to perform the real-time rendering of a game scene. These capabilities are mainly the power consumption and time complexity. The third category is a scenario of

performing parts of the operations on the cloud side while others on the client side. Cognitive Resource Allocation (CRA-GaaS) is the architecture of this cloud gaming scheme. According to system situation, the cloud gaming architecture would be able to combine the processes that can be distributed to execute through the network. The RR-GaaS is considered the most popular and mature model over other models. OnLive and Sony (Gaikai) are ones of the most popular and leading cloud gaming companies that use RR-GaaS.

Many benefits are expected from using RR-GaaS cloud gaming for both developers and players [10]. Unfortunately, many drawbacks are still limiting this approach from being the optimal solution. Computational complexity is one of the main drawbacks. That is a logical reason because of the huge amount of processing that need to be done on the cloud. Running the game, updating the scene, controlling and encoding are all time-consuming processes that are required from the cloud to be achieved in real time.

As a gaming cloud serves huge number of players, any minor reduction in the performance will be raised to be huge and serious problem when multiplied by thousands of real-time players at the same time. Vitally, the encoding time reduction will help reducing the processing and responding delay and power consumption. Encoding time reduction is crucial as it will not only affect processing time itself, but also will reduce clock frequency which decreases power consumption [11].

Several scenarios have been proposed to improve the computational cost. In [12], the complex computational steps have been categorized in four smaller categories: packetization, format conversion, video encoding and memory copy. This research showed that up to 52% of the game processing time is from the video encoding process.

In [13], an investigation of speeding up the cloud gaming operations at the server side has been conducted. The mechanism focused mainly on accelerating power intensive process of video encoding using any available information about a game object on the game engine. The results

showed the motion estimation is the most time-consuming process and the most power wasting step. Applying recent video codecs such as H.264 lead to about 8.86% acceleration of the whole gaming process.

2. RELATED WORKS

The computational complexity is severely challenging the RR-GaaS architecture of game clouding. Games are developed every day in all sides: graphics, actions and resolution. Each one of these factors plays an important key on rendering and encoding time increase. However, several works have been done on scenarios to reduce the encoding time for H.264/AVC and HEVC video codecs. Some other works focused on the computational complexity of the cloud gaming system and how to maintain it at the minimum. As the core of this research is to devise an improvement to the cloud gaming system, this section will present these works and summarize their main ideas.

The real-time video encoding is one challenge since the H.264/AVC has been introduced. Since most games in the market nowadays are online gaming or require live streaming, encoding, decoding and power consumption were monitored and researched to be improved. In [13-18], ideas on reducing the computational complexity of H.264/AVC and HEVC were suggested. The ideas in these works focused on two main steps, these steps can be summarized into: motion estimation and mode decision. They claimed that these two factors affect the overall computational time severely. Some of these research techniques have been suggested to reduce the number of motion estimation search and then save the total encoding time. Other part of these works focused on the mode decision in different conditions to achieve significant speeding up of the encoding total time, also the computational complexity of real-time applications using H.264/AVC encoding system. Final group of the above researches have devised methods by adjusting encoding parameters to adapt the computational complexity of the encoder to the available resources.

These researches have proposed mainly to reduce the computational complexity of video encoding process in general. They expected that will automatically reduce the computational complexity of the cloud gaming overall. Although this is correct somehow, it is still not able to consider the special characteristics of the cloud gaming. For example, they did not show any concern about using the side information that normally come with compressed bitstream in games. Better solutions are more likely to consider all encoded information from the streamed and gained from game scenes.

The computational complexity of cloud gaming has been considered in several researches. Using run-time graphics rendering context, the real-time video encoding process has been monitored and analyzed in [19]. The idea based mainly on selecting set key-frames and exploited the 3-D Warping algorithm to interpolate internal non-key frames. In [20], the depth map from the game engine has been used to avoid the motion estimation process. In [21], the depth information has been also used but this time to increase the motion selection step of the codec H.264/AVC. The authors of [22] proposed an idea to adjust parameters based on the cloud capability. This will exploit the available complexity to meet the complexity provided by the cloud gaming. In [23], authors devised a prioritization method to reduce the power needed for processing. The main idea based on removing objects with less importance to the scene.

3. PROPOSED METHOD

In this paper, a new idea was devised to exploit the side information from the game engine and either skip or process it based on some critical conditions. The proposed idea is more comprehensive than other researches as it considers the macroblock location. Both foreground and background macroblocks would be processed and performed based on its location. For each macroblock in a foreground object, the object movement is detected and compared to the predicted motion vector. As a result of this prediction, three scenarios can be

achieved: skip the motion estimation process, perform the motion estimation process with some restrictions, or perform the normal motion estimation process. The background macroblocks are also considered in this research. The motion estimation for these macroblocks has been accelerated. A background motion vector is used for comparison with the motion vector that already predicted. The three scenarios of motion estimation process will also be applied here for the difference between the background motion vectors and the predicted ones.

The conventional block diagram cloud gaming system relays on performing the gaming operations on both sides, the client and the cloud. The client side is responsible for only collecting the user interactions and decoding the received bitstream from the server side. However, the server side is responsible to run the game logic, render the game objects and deal with frame by frame encoding with the video encoder. All these computational tasks are to be done on the server side. In such cases, the power consumption is a huge challenge on both the cost and ability if the game provider deals with huge number of players. One more challenging in such cases, the server side performs the compression process without any information about the game objects. Although some side information about objects sizes and locations are available, it is not exploited to facilitate the compression operations. Somehow, this information can be used to skip some of the complex encoding operations.

Considering the observations above, we propose a new technique to exploit the object's information available at the encoder side. The main goal is to reduce the computational time with minimum bitrate increase and as low as of the quality degradation. To achieve the goal, two modifications have been introduced. A fast H.264 AVEC encoding proposed in [17] has been exploited to speed up the process. In addition, two paths have been used to detect and improve the encoding process, foreground and background MBs both are considered in the study. The enhanced H.264 codec used is fully describe in [17], however, the paths are described in the following subsections.

3.1 Foreground Macroblocks

As an initial step, each MB location is checked. The MBs that are located inside the border of any game object is considered as foreground macroblocks. After a macroblock is classified as a foreground MB, its movement is checked to examine if its less than the Low Threshold (L-Th), if so, it will be classified as close MB. However, if it is more than L-Th, it would be again examined if its less than the High Threshold (H-Th), if so, it will be classified as average. If the movement is above the H-Th, the MB will be classified as far.

The next stage is to run the motion estimation (ME) for the examined MB. The ME of each MB is skipped if its classified as close. However, the MB will be going through a refinement ME if it is classified as average. Finally, if the MB is classified as far, that means the MB needs to go through the normal ME process.

3.2 Background Macroblocks

The MB is considered as a background MB if it is not located inside any borders of any game objects. As a worst case, the game engine would not provide any information about the background nor its movement. Our proposed algorithm is designed to deal with such cases. However, if the game engine provides the required information, the algorithm is adapted to exploit this information. Although, this will improve the overall processing time, the proposed algorithm can deal with no game engine's feedback using a novel technique.

Initially, the background MB that is currently in process needs to be checked if it is also located in the background of the reference frame. In addition, the current MB is checked if it was located behind any other objects in the previous encoded frame. After checking the previous criteria, if the MB is not a background MB in the previous frame, the normal ME is conducted as full search method. Otherwise, a ME for the current MB is performed based on the background ME with considering the motion vectors for all MBs. In case of an early MBs in a frame, where no motion vectors to be considered, the normal ME will be used. After few MBs, motion

vectors will be extracted and used through similar steps as in foreground MBs.

Figure 1 shows the steps that our algorithm uses at the video encoder side. This process is performed for each MB and for all available modes of each MB. Based on the case of a MB, the ME can be skipped for some modes but performed for another. The first step in the proposed algorithm is to check

if the MB is located inside a game object border, if so, another test is conducted to check in which threshold the MB is classified. The ME is chosen based on the level of the threshold. However, if the MB is located inside a previous frame's background and not in a game object, two different thresholds are examined, and the ME is chosen accordingly. Finally, if the MB is not considered in any of the two previous cases, a normal ME is conducted.

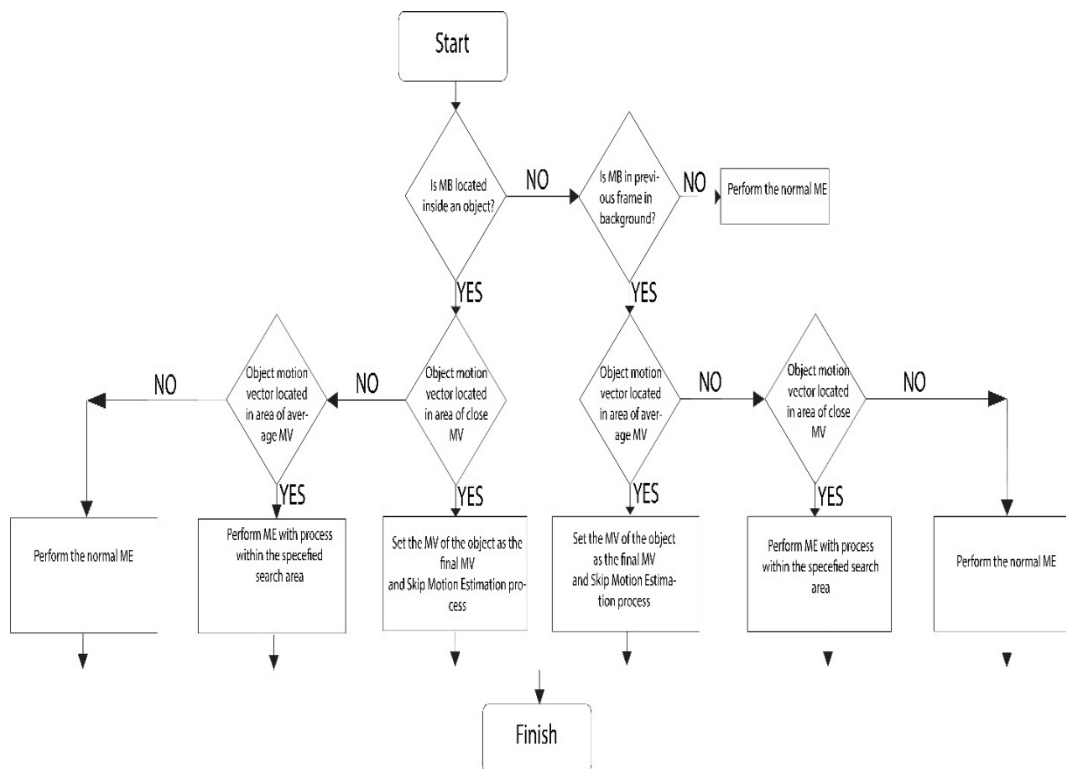


Figure 1 Flowchart of the proposed algorithm

One important factor that can play an essential role in the outcome of the proposed algorithm, is the value of the thresholds. However, these values can be adjusted in initial steps for any specific encoder and depend on the video type or the desired output. The values of the threshold parameters are discussed in section 4, we proposed and used these parameters for the experiments we have done using the H.264 codec.

4. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, several experiments were performed. The H.264 video codec was chosen as a reference software. The experiments considered several types of video sequences that cover different bitrates and resolutions. The chosen parameters of the H.264 codec are shown in Table 1.

Table 1 Parameter Setting Of The Encoder

Profile	Baseline
Level	3
Number of coded frames	100

Number of reference frames	3
Search range	256
RDO	On
Quantization Parameters	28, 30, 32, 34 and 36

For fairness, we conducted the tests and reported the amount of obtained acceleration compared to a state-of-the-art research [13]. On the other hand, we addressed the negative effects of the proposed algorithm on the bitrate increase and quality loss sides. The subjective and objective quality metrics were used to ensure the improvement obtained by the proposed method.

4.1 Subjective Measurements

The subjective quality assessment has been performed based on the Double Stimulus Continuous Quality Scale Method, ITU-R [24]. The scenarios for subjective and objective methods are used. The users' demographic, the methodology and evaluation of tests are also discussed.

Three games were used for evaluation, Aquarium Toy, Death Ball Toy and Truck Toy. All games were tested for both scenarios, the conventional method using H.264 and our proposed method using the improved H.264 [17]. All necessary game's information and encoding settings have been chosen, evaluated, and sent to the video encoder. In addition, the background is also considered and sent to the video encoder, however, no further information is provided by the game engine. In term of capturing the game video, the FRAPS software was used as game scene capturer [25]. The capturing criteria were performed using 30fps and quarter pixel motion estimation. For video encoding, the reference software of H.264 was used with the determined thresholds mentioned above.

Firstly, we asked 20 users as participants in the study. The participants were students at college levels and they have no experience in video compression or assessment. The participants were asked questions to categorize them based on the

demography, the demographic analysis is represented in Table 2.

Table 2 Demographics Of Subjective Participants

Gaming experience				
Bad	Poor	Fair	Good	Excellent
8	9	33	33	17
Monthly game play				
≤5	6-10	11-20	21-30	>30
33	25	26	8	8
Gaming platform (already played on)				
PC	Console	Tablet	Cell phone	
92	42	58	75	

Secondly, we asked participants to score their evaluations on nine different sets of video sequences from the three selected games. Each game has been encoded with three different bitrates as shown in Table3.

Table 3 Selected Bitrates For MOS

Game Name	Rate 1 (Kbps)	Rate 2 (Kbps)	Rate 3 (Kbps)
AquariumToy	1600	1100	800
DeathBallToy	3200	2400	1600
TruckToy	2700	2200	1700

Each set of coded sequences have been coded using the traditional encoding method and our proposed method. To make the results more accurate, we encoded a 60 second video sequence with 1800 frames for each bitstream. Participants must rate the quality as Excellent, Good, Fair, Poor or Bad. They must score in each set of video sequences. The video sequences have been displayed at their original resolution to eliminate any possible distortion that may occur as a result of any scaling. The participants' viewing distance was determined to be four times than the screen size as recommended in [24].

Finally, the Mean Opinion Score (MOS) was used as the subjective quality metric for both the conventional and proposed methods. The MOS for each combination methods and setting is calculated

as an average of all MOS of all participants. The results of all MOS are shown in Table 4. As its

clearly shown, the MOS of the proposed method is about 9.23% less than the conventional coding one.

Table 4 Subjective results - MOS

	AquariumToy			DeathBallToy			TruckToy			Average Change (%)
	Rate 1	Rate 2	Rate 3	Rate 1	Rate 2	Rate 3	Rate 1	Rate 2	Rate 3	
Conventional	4.71	4.52	4.97	4.89	4.76	4.53	4.85	4.63	4.69	-
Proposed	3.71	3.86	3.31	3.99	4.01	4.07	4.09	4.07	3.99	9.23

Moreover, the results of the proposed method showed consistent performance over the three different rates chosen for the experiments. The average quality of the proposed method degrades negligibly by less than 0.3 dB compared to the conventional full method. Both the PSNR and the dB are an evidence on the high performance of the proposed method. As the motion estimation process is mainly performed for finding the best block, the entropy coding is used to calculate the residual between current and reference block.

4.2 Objective Measurements

The conventional compression method is illustrated in Figure 2 and the proposed method is illustrated in Figure 3. The Torque2D is described in [23], this method is used as the testing game engine. The same three games as in subjective metric also used for the objective testing. For each game, an object has been chosen and sent to the video encoder. In addition to the selected object, the background is taken into consideration for speeding up. However, the information about the background is not provided by the game engine. The game scenes are captured in order to be sent to the encoder by FRAPS software [25]. The capturing frame rate was set to 30fps. The H.264 reference software of full search is used for video encoding. To conduct the experiments, the thresholds are defined as follows: T1=20, T2=10, T3=2 and T4=10. The mentioned thresholds are to be determined; an initial simulation process needs to be conducted first on a set of video sequences outside the specified evaluation pool. The simulation process should contain vary contents and bitrates. The thresholds for any other video codecs need to be adjusted using

the initial simulation process again. However, the threshold parameters are not changeable after the simulation process. For performance, the proposed method are evaluated and tested in different settings. Table 1 shows the parameter settings that used in the video encoder in order to conduct the evaluation performance. The captured videos are coded using Baseline with five different QP values and 100 coded frames.

The proposed method showed significant speed up compared to the conventional scheme. The total encoding time for the three selected video scenes are shown in Figure 4. As it illustrated in the figure, the total encoding time after implementing the proposed method is speeded up by up to 18% over the selected QPs. The significant speeding up of the ME for the proposed method is shown in Figure 5. As its clearly illustrated, the speed of the ME is increased by up to 42% and significantly outperformed the conventional encoding method.

The Rate-Distortion (RD) performance for the proposed method, conventional method and the comparison are shown in Figure 6. Greatly, the proposed method has achieved a close RD performance to the conventional method with negligible amount of increment. The proposed method adopted a technique to reduce the quality degradation occurred as a result of the miss-prediction. As a conclusion, losing less than 0.1 PSNR with achieving up to 18% speeding up compared to the conventional method should be a negligible issue.

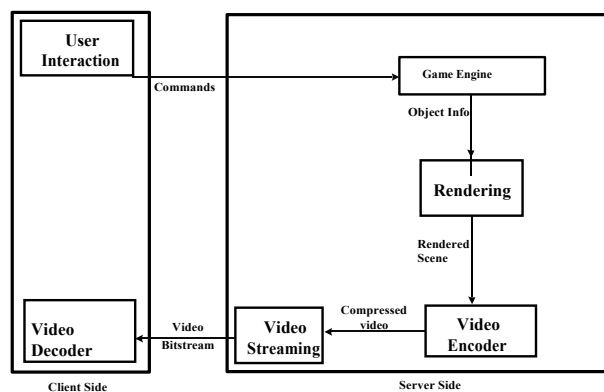


Figure 2 Conventional Compression Method

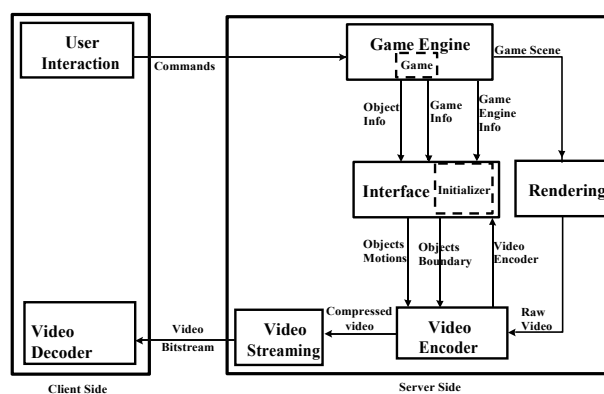


Figure 3 Diagram of the Proposed Method

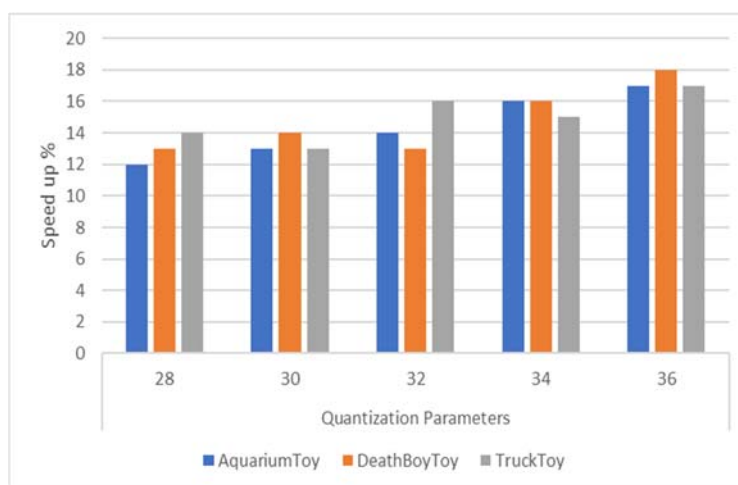


Figure 4 Encoding Time Speedup Compared to the Conventional Method

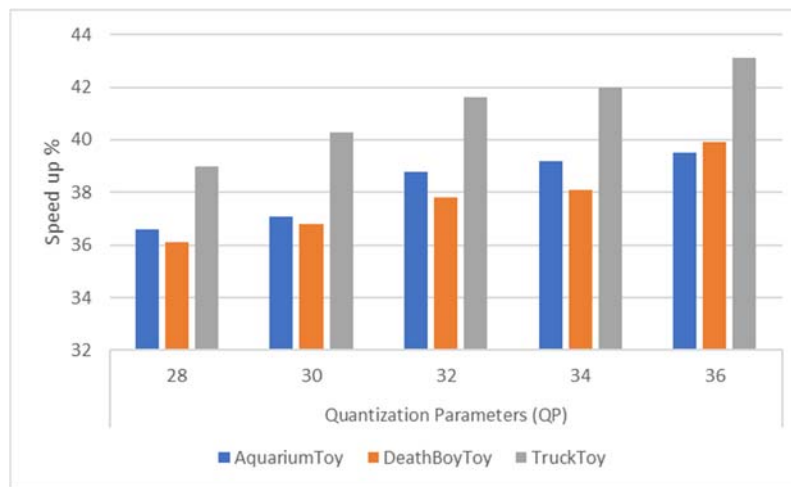


Figure 5 Motion Estimation Compared to the Conventional Method

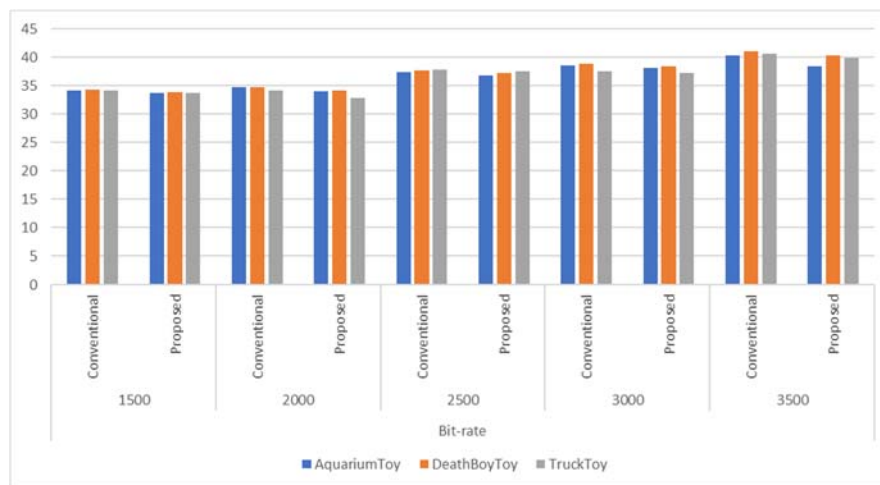


Figure 6 Rate Distortion Performance Compared to the Conventional Method

5. CONCLUSION

A new accelerated cloud gaming encoding technique was proposed in this research. The information of a game object inside the game engine was exploited to achieve this acceleration. In particular situations, this game object information was used in the video encoder to skip the time-consuming motion estimation process. In addition, the macroblocks are classified either as foreground or background macroblocks. In case of a macroblock is considered as a background one, some criteria to be checked, if met, the motion estimation is accelerated and the quality

improvement also achieved. The experiments showed that the proposed method is able to speed up the motion estimation by up to 43% and the encoding time up to 18%. One big benefit of the proposed method is the capability of implementing it inside any game object and with any video encoder with minimal required modifications. However, supporting different type of game objects such as 3D objects may need more investigations and could be one of the future work areas.

REFERENCES

- [1] "Industry Facts," Entertainment Software Association (ESA). [Online]. Available: <http://www.theesa.com/facts/index.asp>.
- [2] "Gartner's Press Release," Gartner Inc. [Online]. Available: <http://www.gartner.com/newsroom/id/261491>.
- [3] Mazhar, A.: Efficient Cloud Gaming Scheme Using Scene Objects Adaptation. International Journal of Computer Science and Information Technology (IJCSIT), Vol. 9, No 2. 2017
- [4] W. Cai, H. Chan, X.Wang, And V. Leung, "Cognitive Resource Optimization For The Decomposed Cloud Gaming Platform," Ieee Trans. Circuits Syst. Video Technol., Vol. 25, No. 12, Pp. 2038–2051, Dec. 2015.
- [5] D. Mishra, M. El Zarki, A. Erbad, C.-H. Hsu, And N. Venkatasubramanian, "Clouds + Games: A Multifaceted Approach," Ieee Internet Comput., Vol. 18, No. 3, Pp. 20–27, May 2014.
- [6] S.-P. Chuah, C. Yuen, And N.-M. Cheung, "Cloud Gaming: A Green Solution To Massive Multiplayer Online Games," Ieee Wireless Commun., Vol. 21, No. 4, Pp. 78–87, Aug. 2014.
- [7] W. Cai "Survey on Cloud Gaming: Future of Computer Games" 2169-3536 2016 IEEE. Volume 4 2016.
- [8] Yeng-Ting Lee, Kuan-Ta Chen, Yun-Maw Cheng, and Chin-Laung Lei, "World of Warcraft Avatar History Dataset" In Proceedings of ACM Multimedia Systems, 2011.
- [9] W. Cai, M. Chen, and V. Leung, "Toward Gaming as a Service," IEEE Internet Comput., vol. 18, no. 3, pp. 12–18, 2014.
- [10] R. Shea, J. Liu, E. Ngai, and Y. Cui, "Cloud gaming: architecture and performance," IEEE Netw., pp. 1–23, 2013.
- [11] T. D. Burd and R. W. Brodersen, "Processor Design for Portable Systems," J. VLSI Signal Process. Syst., vol. 13, no. 2–3, pp. 203–221, 1996.
- [12] C.-Y. Huang, K.-T. Chen, D.-Y. Chen, H.-J. Hsu, and C.-H. Hsu, "GamingAnywhere: The first open source cloud gaming system," ACM Trans. Multimed. Comput. Commun. Appl., vol. 10, no. 1s, 2014.
- [13] M. Semsarzadeh, M. Hemmati, A. Javadtalab, A. Yassine, and S. Shirmohammadi, "A Video Encoding Speed-up Architecture for Cloud Gaming," in Cloud Gaming Systems and Networks in conjunction with IEEE International Conference on Multimedia & Expo, 2014.
- [14] M. Khan, M. Shafique, M. Grellert, and H. Jörg, "Hardware-software collaborative complexity reduction scheme for the emerging HEVC intra encoder," in Design, Automation and Test in Europe (DATE), 2013, pp. 125–128.
- [15] J. Zhou, D. Zhou, and S. Goto, "Alternating asymmetric search range assignment for bidirectional motion estimation in H.265/HEVC and H.264/AVC," J. Vis. Commun. Image Represent., vol. 25, no. 5, pp. 1275–1286, Jul. 2014.
- [16] W. Lin, K. Panusopone, D. M. Baylon, M. Sun, and Z. Chen, "A Fast Sub-Pixel Motion Estimation Algorithm for H.264/AVC Video Coding," IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 2, pp. 237–242, 2011.
- [17] Joint reference frame inter-mode selection for fast H. 264 video coding, AA Mazhar, AM Abdalla, "Signal, Image And Video Processing", 10 (4), 617-623, 2016.
- [18] Performance evaluation of H. 265/MPEG-HEVC VP9 and H. 264/MPEG AVC video coding, A Mazhar, "The International Journal of Multimedia & Its Applications" (IJMA) 8 (1), 35-44, 2016.
- [19] S. Shi, C. Hsu, K. Nahrstedt, and R. Campbell, "Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming," in ACM Multimedia, 2011, pp. 103–112.
- [20] N. Tizon, C. Moreno, and M. Preda, "ROI based video streaming for 3D remote rendering," in IEEE International Workshop on Multimedia Signal Processing (MMSP), 2011, pp. 1–6.

- [21] G. Cheung, A. Ortega, and T. Sakamoto, “Fast H. 264 mode selection using depth information for distributed game viewing,” *Vis. Commun. Image Process.*, 2008.
- [22] M. R. Hosseinzadeh Taher, H. Ahmadi, and M. R. Hashemi, “Power-Aware Analysis of H.264/AVC Encoding Parameters for Cloud Gaming,” in *Cloud Gaming Systems and Networks in conjunction with IEEE International Conference on Multimedia & Expo*, 2014.
- [23] M. Hemmati, A. Javadtalab, A. A. Nazari Shirehjini, S. Shirmohammadi, and T. Arici, “Game as video: Bit rate reduction through adaptive object encoding,” in *ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2013, pp. 7–12.
- [24] ITU-R, “Methodology for the subjective assessment of the quality of television pictures”.
- [25] “FRAPS, Real-time video capture & benchmarking.” [Online]. Available: <http://www.fraps.com>.
- [26] “Torque2D Game Engine.” [Online]. Available: <http://www.garagegames.com/products/torque-2d>.