Remote VR Gaming on Mobile Devices

Mikko Pitkänen, Marko Viitanen, Alexandre Mercat and Jarno Vanne

Tampere University
Tampere, Finland
{mikko.pitkanen, marko.viitanen, alexandre.mercat, jarno.vanne}@tuni.fi

ABSTRACT

This paper presents a remote 360-degree virtual reality (VR) gaming system for mobile devices. In this end-to-end scheme, execution of VR game is off-loaded from low-power mobile devices to a remote server where the executed game is rendered based on controller orientation and actions transmitted over the network. The server is running the Unity game engine and Kvazaar video encoder. Kvazaar compresses the rendered views of the game to *High Efficiency Video Coding* (*HEVC*) video that is streamed to a player over a regular WiFi link in real time. The frontend of our proof-of-concept demonstrator setup is composed of the Samsung Galaxy S8 smartphone and Google Daydream View VR headset with a controller. The backend server is a laptop equipped with Nvidia GTX 1070 GPU and Intel i7 7820HK CPU. The system is able to run the demonstrated 360-degree shooting VR game with 1080p resolution at 30 fps while keeping motion-to-photon latency close to 50 ms. This approach lets players enter immersive gaming experience without a need to invest in all-in-one VR headsets.

CCS CONCEPTS

• Computing methodologies → Virtual reality • Computing methodologies → Image compression

KEYWORDS

Virtual Reality (VR), 360 video, Video encoding, High Efficiency Video Coding (HEVC), Remote gaming

ACM Reference format:

Mikko Pitkänen, Marko Viitanen, Alexandre Mercat, and Jarno Vanne. 2019. Remote VR Gaming on Mobile Devices. In *Proceedings of ACM Multimedia conference (ACMMM'19). ACM, Nice, France, 2 pages.* https://doi.org/10.1145/3343031.3350595

1 Introduction

In recent years, *virtual reality* (*VR*) gaming has seen a tremendous increase in the number of VR-ready devices, widespread games, and active users. According to Cisco [1], the global IP traffic for virtual and augmented reality applications will grow 12-fold from 2017 to 2022 and the corresponding increment in Internet remote gaming will be 9-fold. The growth will be fastest in mobile platforms due to their cost-effective ability to exploit mobile phones as *head mounted device* (*HMD*) screens.

Even though mobile devices do not offer the same graphical performance as the high-end desktop computers, their wireless networking capabilities and built-in hardware video decoders make it possible to render a game on a remote server and use video for communication between the server and a mobile client. The latest video coding standard, *High Efficiency Video Coding* (*HEVC/H.265*) [2], reduces the bit rate by almost 40% over its predecessor *Advanced Video Coding (AVC/H.264)* [3]. The decoding hardware for HEVC is available in the latest smartphones, so HEVC can be considered a potential technology enabler for remote mobile gaming even within the limits of the current network capacities.

The main limitation of this approach comes from the computational performance and the bounded energy density of batteries in mobile devices. The recent progress made in microelectronics of the embedded multi-core and GPU platforms is incapable of compensating for the increasing computational complexity of multimedia applications. Hence, maximizing *Quality of Experience* (*QoE*) in VR gaming and video playback represents a serious challenge to low-power mobile devices.

Nowadays, a few remote gaming solutions featuring compressed video exist in the market, such as the commercial Amazon AWS cloud servers [4] and open-source approaches by [5], [6], but they do not address the VR gaming aspect. Authors in [7] investigated three possible solutions for high bit rate and ultra-low latency VR/AR streaming: *Field of View* (*FoV*) video, 360-degree video, and models for *6-Degrees of Freedom* (*6DoF*) video. They also concluded that further performance improvement could be achieved by replacing AVC by HEVC encoder. Skupin et al. [8] proposed a similar approach for FoV for non-interactive video content using adaptive HEVC video system.

In this work, we propose a real-time, interactive remote 360-degree VR gaming framework for low-power mobile devices. To the best of our knowledge, this is the first embedded solution with a working prototype in the field. The server end utilizes Unity® game engine to generate raw VR video which is encoded in real-time by an award-winning Kvazaar open-source HEVC encoder [9]. Our solution exploits the hardware HEVC decoder of the chosen embedded system to achieve real-time performance.

The rest of the paper is organized as follows. Section 2 gives an in-depth overview of the server and client ends in the proposed system. Section 3 presents the demonstration setup and visitor experience. Section 4 concludes the paper.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10.

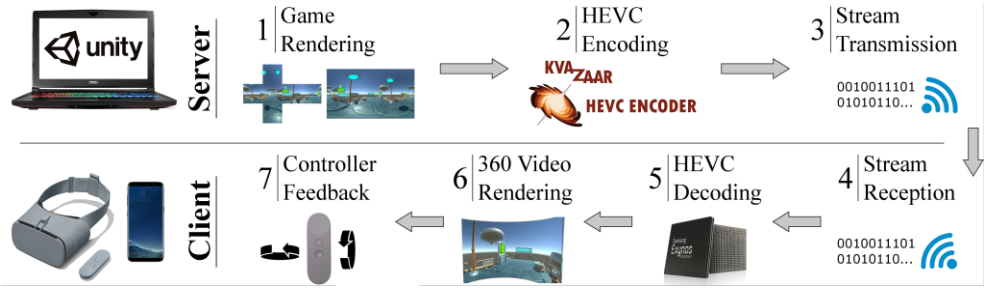

Figure 1: Overview of the proposed client-server system divided into seven stages.

2 Proposed Remote VR Gaming System

Figure 1 illustrates an overview of the proposed client-server gaming architecture split into seven stages. This proposal is a direct continuation of our previous work [10]. The novelty of this contribution lies in replacing high-end laptop clients of [10] with low-power mobile phones.

2.1 Server

The server includes only standard CPUs and GPUs with no dedicated hardware accelerators. It runs a game built with the Unity game engine. The game, originally designed for HTC Vive VR headset, is adapted for Google Daydream View headset and controller in this demonstration. The server process is divided into three stages described below and in Figure 1:

1) **Game rendering.** A camera rig of six virtual cameras is set to render a cube map composed of top, bottom, and four side views of 960×960 pixels. The cube map is processed with an integrated stitching, *equirectangular projection* (*ERP*), and RGB to YUV conversion shader which outputs a video frame in a rectangular YUV420 1920×1080 (1080p) format. The output frame is transferred from GPU memory to CPU memory using OpenGL native plugin.
2) **HEVC encoding**. The raw ERP video is intra encoded into HEVC format with Kvazaar, which is shown to be the leading open-source HEVC intra encoder [11].
3) **Stream transmission.** The HEVC stream is transmitted from the server to the client over the wireless network using the *Transmission Control Protocol* (*TCP*).

The first stage is executed on a GPU and the other two on a CPU.

2.2 Client

The main novelty of this work lies in using a mobile phone as a client. The client is running a custom-written Android application that uses the Google VR SDK and OpenGL ES which is dedicated to embedded systems. The client process is split into four steps described next and in Figure 1:

4) **Stream reception.** The application is invoked by establishing the TCP connection between the server and client. The server sends encoding parameters to the client after the connection has been established. The client receives the video stream on a frame-by-frame basis.
5) **HEVC decoding.** The client decodes each frame using the built-in HEVC *System on Chip* (*SoC*) decoder which is enabled by the Android MediaCodec API.
6) **360 video rendering**. The decoded frame is loaded to an OpenGL texture which is projected on a sphere constructed around the user's virtual position to render the 360 video.
7) **Controller feedback.** The client sends controller feedback through the TCP connection and the server updates the state of the game accordingly. The feedback is composed of the controller orientation and the button states.

The stage 5 is implemented on a dedicated hardware accelerator, stage 6 on a GPU, and the other two stages on a CPU.

3 Demonstration Setup

The proposed demonstration setup is composed of a gaming grade laptop and a smartphone. The laptop acts as a server and it is equipped with Nvidia GTX 1070 GPU and Intel i7 7820HK CPU running the Windows 10 operating system. A Samsung Galaxy S8 smartphone acts as a client and it features a 5.8" Quad HD+ Super AMOLED (2960×1440) display, octa-core Exynos 8895 SoC, 4 GB of RAM, and the Android 9.0 operating system. It is combined with a Google Daydream View headset and controller for the interactive VR demonstration. The laptop and smartphone are connected via a regular TCP over WiFi connection.

During the demonstration, visitors will play a VR shooting game through the Google Daydream View. In the game, a player controls a stationary turret with a controller and shoots at targets by pressing a button on the controller.

The system is able to run the demonstrated game with 1080p resolution at 30 *frames per second* (*fps*) on both server and client ends. The client renders the 360 scene locally enabling an average motion-to-photon latency of around 50 ms. Network usage is 1.2 Mbps on average over the gaming session. The game runs for around three hours on battery and reaches a temperature of 50°C.

4 Conclusion

This paper presented a real-time remote 360 VR gaming system for mobile devices whose performance and power dissipation have to be taken into account to meet QoE goals. Our solution deployed the state-of-the-art HEVC coding technology and built-in hardware accelerators of the mobile SoCs to achieve immersive gaming experience with low latency and affordable price.

ACKNOWLEDGMENTS

This work was supported in part by the European Celtic-Plus Project VIRTUOSE and the Academy of Finland (decision no. 301820).

REFERENCES

[1] Cisco, Cisco Visual Networking Index: Forecast and Trends, 2017-2022, Feb. 2019.

[2] *High Efficiency Video Coding*, document ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, Apr. 2013.

[3] *Advanced Video Coding for Generic Audiovisual Services*, document ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC, Mar. 2009.

[4] Parsec [Online]. Available: https://parsecgaming.com

[5] J. Beyer and R. Varbelow, "Stream-A-Game: an open-source mobile cloud gaming platform," *in Proc. Int. Workshop on Network Systems Support for Games*, Zagreb, Croatia, Dec. 2015.

[6] C. Y. Huang, C. H. Hsu, Y. C. Chang, and K. T. Chen, "GamingAnywhere: an open cloud gaming system," *in Proc. ACM Multimedia Syst. Conf.*, Oslo, Norway, Feb. 2013.

[7] X. Hou, Y. Lu, and S. Dey, "Wireless VR/AR with edge/cloud computing," *in Proc. IEEE Int. Conf. Computer Commun. Networks*, Vancouver, Canada, July 2017.

[8] R. Skupin, Y. Sanchez, C. Hellge, and T. Schierl, "Tile based HEVC video for head mounted displays," *in Proc. IEEE Int. Symp. Multimedia*, San Jose, California, USA, Dec. 2016.

[9] M. Viitanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vanne, and T. D. Hämäläinen, "Kvazaar: open-source HEVC/H.265 encoder," *in Proc. ACM Int. Conf. Multimedia,* Amsterdam, The Netherlands, Oct. 2016. DOI: https://doi.org/10.1145/2964284.2973796

[10] M. Viitanen, J. Vanne, T. D. Hämäläinen, and A. Kulmala, "Low latency edge rendering scheme for interactive 360 degree virtual reality gaming," *in Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Vienna, Austria, Jul. 2018.

[11] A. Mercat, A. Lemmetti, M. Viitanen, and J. Vanne, "Acceleration of Kvazaar HEVC intra encoder with machine learning," *in Proc. IEEE Int. Conf. on Image Processing*, Taipei, Taiwan, Sep. 2019.