

# Kvazaar HEVC Still Image Coding on Raspberry Pi 2 for Low-Cost Remote Surveillance

Marko Viitanen, Ari Koivula, Jarno Vanne, Timo D. Hämäläinen

Department of Pervasive Computing, Tampere University of Technology, Finland

{marko.viitanen, ari.koivula, jarno.vanne, timo.d.hamalainen}@tut.fi

**Abstract**—This demonstrator serves as a proof-of-concept of our multi-camera remote surveillance system that supports 1080p still image capture with a 10-second refresh rate. Image capture, compression, and broadcast are implemented in each Raspberry Pi 2 camera node. Image compression is conducted with an open-source Kvazaar HEVC encoder that outputs HEVC images in BPG format. The BPG images are broadcast from camera nodes to terminals over the Internet through WebSocket protocol. The images can be played back with most Web browsers in remote locations with Internet access.

**Index Terms**— High Efficiency Video Coding (HEVC), Kvazaar HEVC encoder, Raspberry Pi 2, WebSocket, Better Portable Graphics (BPG) image format, remote surveillance

## I. INTRODUCTION

Implementing a remote surveillance system in rural areas tends to need affordable camera devices accessible over low-bit rate connections. In our proposal, each camera device is implemented on a fully-equipped Raspberry Pi 2 Model B [1] whose current price is below 70\$. Low bit rate target is addressed by Kvazaar HEVC encoder [2] that is the best practical open-source encoder for efficient intra picture compression [3]. In this work, Kvazaar has been extended to output still images in a *Better Portable Graphics (BPG)* format [4] supported by most Web browsers through a small JavaScript BPG decoder. The communication between camera devices and terminals is managed with WebSockets.

## II. DEMONSTRATION SETUP

The proposed remote surveillance system can support arbitrary number of camera nodes (servers) and terminals (clients). Fig. 1(a) shows an end-to-end data flow diagram of the implemented client-server model. It involves still image 1) capture; 2) encoding; 3) broadcast; and 4) playback.

In the server, the images are taken by Raspberry Pi camera module. A command line tool called *raspiyuv* is used to acquire a raw image from the camera to Kvazaar for encoding.

Raspberry Pi contains a 900 MHz quad-core ARM Cortex-A7 processor on which Kvazaar is multithreaded with an HEVC data-level parallelization technique called *tiles* [5] that split a picture into independent regions for parallel encoding. Kvazaar outputs the encoded image in a BPG format [4].

The server broadcasts BPG images to the clients. The server end is built with *Node.js*. Each client opens an HTTP connection to download a JavaScript program that handles the WebSocket connection with the server.

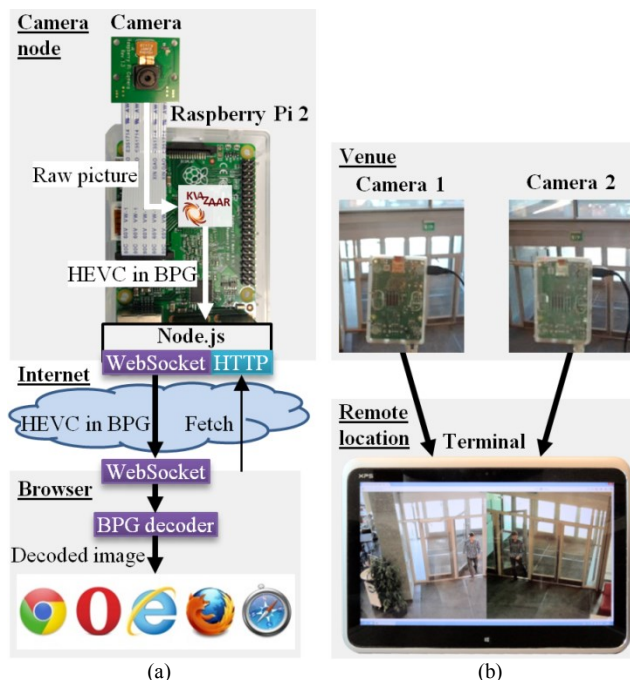


Fig. 1. Demonstrator. (a) Block diagram. (b) Surveillance with two cameras.

The clients receive BPG images from the server through full-duplex communication via WebSockets without polling. Since WebSocket protocol hides any data fragmentation from the user, the BPG images can be immediately decoded with the JavaScript decoder. Finally, new images are rendered on a HTML5 Canvas as soon as they are decoded.

Fig. 1(b) illustrates a case study where two indoor cameras are used to monitor the entrance to the building. The cameras capture 1080p images every 10 seconds from the left and right side of the entrance and send them over the Internet to the laptop screen in the remote location.

## REFERENCES

- [1] *Raspberry Pi 2 Model B* [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
- [2] *Kvazaar HEVC encoder* [Online]. Available: <https://github.com/ultravideo/kvazaar>
- [3] M. Viitanen, A. Koivula, A. Lemmetti, J. Vanne, and T. D. Hämäläinen, "Kvazaar HEVC encoder for efficient intra coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, Lisbon, Portugal, May 2015.
- [4] *BPG Image Format* [Online]. Available: <http://bellard.org/bpg/>
- [5] *High Efficiency Video Coding*, document ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, Apr. 2013.
- [6] *The WebSocket Protocol* [Online]. Available: <http://tools.ietf.org/html/rfc6455>