

# Live Demonstration: Run-time Visualization of Kvazaar HEVC Intra Encoder

Marko Viitanen, Ari Koivula, Jarno Vanne, and 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 presents a run-time visualization tool for Kvazaar HEVC intra encoder. The implemented open-source tool is seamlessly integrated into Kvazaar to provide instant visual feedback of the encoding process. The visualization overlays the reconstructed HEVC video with the boundaries of the used block partitioning structure and associated intra prediction modes. The tool is also able to illustrate all Kvazaar parallelization schemes at run time: Wavefront Parallel Processing, tiles, and picture-level parallel processing. The displayed visualization information can be gradually adjusted to user needs. The tool is primarily designed for Kvazaar debugging but it also suits educational purposes.

**Keywords**— *High Efficiency Video Coding (HEVC); Kvazaar HEVC encoder; Intra prediction modes; Block partitioning structure; visualization*

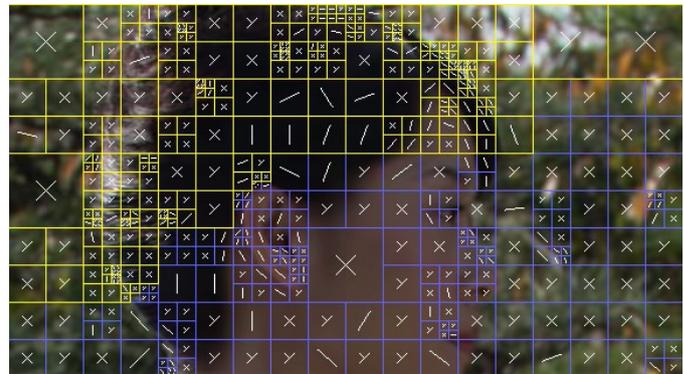
## I. INTRODUCTION

The latest video coding standard *HEVC (High Efficiency Video Coding)* [1] adopts a quad-tree block partitioning structure for intra coding with block sizes of  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$ , and  $64 \times 64$  pixels. Each block size can be predicted by one of 35 *intra prediction (IP)* modes (DC, planar, and 33 angular). To detect and analyze all these combinations in a picture makes visualization an integral part of an encoder development. However, it is rarely included in the encoder itself but a 3<sup>rd</sup> party software is required. Some open-source HEVC visualizers such as *Gitl HEVC/H.265 Analyzer* [2] exist, but they fail to meet demands of real-time debugging by only providing visualization by processing encoded binary file.

This demonstration setup running on a laptop shows the visualization tool implemented directly to the search functions of Kvazaar [3], [4]. This provides real-time visual feedback of the coding process. Visitors will experience the inner workings of a modern encoder. Even the blocks not selected for the final output are shown as they are tested.

## II. FEATURES OF VISUALIZER

The visualizer is built using *Simple DirectMedia Layer (SDL)* multi-platform graphics library version 2. The rendered illustration layers are 1) *block partitioning*; 2) *intra angular*; and 3) *block reconstruction*. They are illustrated in Figure 1. The block reconstruction is used as the background layer and the other two layers can be removed on simple key commands. This way, Kvazaar developers can focus on visualization elements of particular interest. The intra angular layer excludes DC and Planar modes to avoid confusion with the angular modes that can be used as such to detect suboptimal modes.



(a) All visualization layers



(b) Intra angular and block reconstruction layers

Fig. 1. Kvazaar visualization during picture-level parallel processing.

Besides single-threaded encoding, the tool can also visualize Kvazaar parallelization schemes including Wavefront Parallel Processing, tiles, and picture-level parallel processing [4]. Parallel pictures are marked with distinct block border colors. Further information about the tool is available online in [5].

## REFERENCES

- [1] *High Efficiency Video Coding*, document ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, Apr. 2013.
- [2] *Gitl HEVC/H.265 Analyzer* [Online]. Available: <https://github.com/lheric/GitlHEVCAnalyzer>
- [3] *Kvazaar HEVC encoder* [Online]. Available: <https://github.com/ultravideo/kvazaar>
- [4] A. Koivula, M. Viitanen, J. Vanne, T. D. Hämäläinen, and L. Fasnacht, "Parallelization of Kvazaar HEVC intra encoder for multi-core processors," in *Proc. IEEE Workshop Signal Process. Syst.*, Hangzhou, China, Oct. 2015.
- [5] *Kvazaar Visualizer* [Online]. Available: <http://ultravideo.cs.tut.fi/#visualizer>