Low Delay Complexity Constrained Encoding

T. Vermeir, J. Slowack, G. V. Wallendael, P. Lambert, and R. V. d. Walle

In: 2016 Data Compression Conference (DCC), 635-635, 2016.

Low Delay Complexity Constrained Encoding

Thijs Vermeir[*+], Jürgen Slowack[*],
Glenn Van Wallendael[+], Peter Lambert[+], Rik Van de Walle[+]

[*]*Barco N.V.*
Beneluxpark 21
8500 Kortrijk, Belgium
thijs.vermeir@barco.com

[+]*Data Science Lab, Ghent University*
Sint-Pietersnieuwstraat 42
9000 Ghent, Belgium
thijs.vermeir@ugent.be

Complex software applications typically consist of multiple software processes running concurrently to exploit the available computational resources in the hardware. However, the computational complexity of these software processes is often variable and the processes can interfere with each other. This can be an issue for real-time applications with a fixed deadline such as low delay video encoding. This is even more important when using High Efficiency Video Coding (HEVC), since HEVC encoding consumes 9 to 502% more computational complexity compared to its predecessor H.264/AVC.

The state-of-the-art in complexity optimization for HEVC is extensive. However, most algorithms described in the literature do not provide complexity control, but instead focus only on reducing complexity, typically, through early termination of the mode decision or motion search process. With complexity constrained encoding, it is the intent to keep the complexity constant around a predefined complexity target while maximizing the video quality. In this paper, we propose a technique to control the computational complexity of an HEVC encoder. Our technique consists of adaptively selecting either the Merge 2Nx2N mode or full evaluation independent of the CU depth, based on a threshold on the expected RD cost error. This threshold is varied frame-by-frame by a controller, based on the actual frame encoding time. As a result, regardless of the host system and the video sequence, after a short initialization phase, our technique allows constraining the complexity to a preconfigured complexity target. Convergence to this target threshold is fast (e.g., Figure 1) while incurring only a limited loss in RD performance (Figure 2).

Our technique can be used for software encoding applications where optimal usage of available computational complexity is key (e.g. cloud encoding). Possible future improvements to this method include extending the number of encoding mode groups.

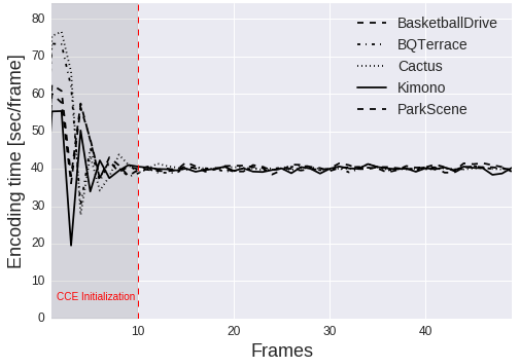

Figure 1: Encoding time per frame with QP=22 for complexity constrained encoder based on HM 16.2 with a complexity target of 40 sec/frame.

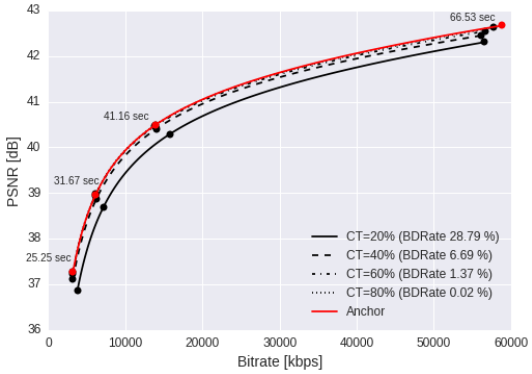

Figure 2: RD curves for the kimono sequence with different complexity targets.