Parallel Implementations of Lambda Domain and R-Lambda Model Rate Control Schemes in a Practical HEVC Encoder

Joose Sainio, Alexandre Mercat, and Jarno Vanne

Ultra Video Group, Tampere University, Finland

{joose.sainio, alexandre.mercat, jarno.vanne}@tuni.fi

This paper summarizes the key parallelization aspects of the two most popular academic *rate control* (*RC*) algorithms for *High Efficiency Video Coding* (*HEVC*): 1) *Lambda domain* (*LD*) [1] and 2) *R-Lambda model* (*R-LM*) [2] that were originally designed for the *HEVC test model* (*HM*). In this work, these two sequential RC algorithms were ported to the practical Kvazaar open-source HEVC encoder [3] and adapted to work with *wavefront parallel processing* (*WPP*) and *overlapping wavefront* (*OWF*) parallelization techniques.

LD does not need to address WPP because the *Coding Tree Units* (*CTUs*) of a frame are independent. However, OWF requires extra synchronization because RC parameters for collocated CUs might be accessed simultaneously.

Conversely, R-LM requires limiting RC parameter reallocation for WPP as otherwise the reallocation would go beyond wavefront boundaries. As for LD, extra synchronization is needed to deal with OWF. Moreover, R-LM uses the parameters of every CTU to provide an initial allocation for each CTU. As the parameters are updated, they will be different for the reallocation which causes noticeable quality degradation. This is alleviated by updating the parameters after the complete frame is encoded, instead of each CTU. All these solutions have either unmeasurable or negligible effect on the encoding speed.

The comparison of LD and R-LM was carried out with 19 natural HEVC CTC sequences on a Ryzen Threadripper 2990WX processor. The thread count was set to 16 in all test runs. The configurations were ultrafast preset with *random access* (*RA*) GOP, ultrafast with *Low Delay B* (*LB*) GOP, veryslow RA GOP, and veryslow LB GOP. Table 1 tabulates PSNR, SSIM, and VMAF BD-rates, speedup, and bitrate error between the algorithms in sequential execution. Table 2 reports the respective results with OWF and WPP. Overall, both algorithms are suited for real-time encoding.

References

[1] B. Li, H. Li, L. Li, and J. Zhang, “λ domain rate control algorithm for High Efficiency Video Coding,” *IEEE Trans. Image Process.,* vol. 23, Sept. 2014, pp. 3841–3854.

[2] S. Li, M. Xu, Z. Wang, and X. Sun, “Optimal bit allocation for CTU level rate control in HEVC,” *IEEE Trans. Circuits Syst. Video Technol.,* vol. 27, Nov. 2017, pp. 2409–2424.

[3] A. Lemmetti, M. Viitanen, A. Mercat, and J. Vanne, “Kvazaar 2.0: fast and efficient open-source HEVC inter encoder,” *in Proc. 11th ACM Multimedia Syst. Conf.*, New York, USA, May 2020, pp. 237–242.

Table 1: Sequential performance of R-LM over LD in Kvazaar.

Encoding configuration	PSNR BD-rate (%)	SSIM BD-rate (%)	VMAF BD-rate (%)	Speed difference	Bitrate error (%)	
					R-LM	***LD***
Ultrafast RA GOP	-2.36	-5.87	-2.52	0.99×	0.95	-0.29
Ultrafast LB GOP	-2.22	-7.07	-2.30	0.98×	0.79	0.22
Veryslow RA GOP	-4.67	-8.89	-3.80	1.05×	-0.22	-0.97
Veryslow LB GOP	-1.46	-3.52	-0.76	1.03×	1.51	0.87

Table 2: Parallelized (WPP+OWF) performance of R-LM over LD in Kvazaar.

Encoding configuration	PSNR BD-rate (%)	SSIM BD-rate (%)	VMAF BD-rate (%)	Speed difference	Bitrate error (%)	
					R-LM	***LD***
Ultrafast RA GOP	-1.88	-5.34	-2.35	0.98×	1.53	0.68
Ultrafast LB GOP	-2.62	-6.82	-2.24	0.98×	0.90	0.35
Veryslow RA GOP	-3.28	-6.73	-1.77	1.03×	0.54	-0.14
Veryslow LB GOP	-2.91	-5.09	-0.87	1.01×	2.07	1.41