COMPLEXITY-CONSTRAINED RATE-DISTORTION OPTIMIZATION FOR H.264/AVC VIDEO CODING

Tiago A. da Fonseca and Ricardo L. de Queiroz

Universidade de Brasilia
Department of Electrical Engineering
Brasilia, DF, Brazil
tiago@image.unb.br, queiroz@ieee.org

ABSTRACT

In order to enable real-time software-based video encoding, in this work we optimized the prediction stage of an H.264 video encoder, in the complexity sense. Thus, besides looking for the coding options which lead to the best coded representation in terms of rate and distortion (RD), we constrain to a complexity (C) budget. We present a complexity optimized framework (RDC-optimized) which allows for real-time video compression and that does not make use of frame-skipping to comply to the desired encoding speed. We developed our framework around an open source software implementation of the H.264/AVC, the the x264 encoder. Results show that tight complexity control is attainable in practice, with very little loss in RD performance.

***Index Terms*—** H.264, mode decision, complexity scalability, rate control, real-time video coding.

1. INTRODUCTION

The H.264/AVC standard [1] results from the integration of many modern digital signal processing techniques to deliver enhanced code efficiency for applications like video telephony, video conferencing, digital TV, streaming video etc. The H.264/AVC coder has been well decribed in the literature [2, 3, 4, 5].

As a hybrid DPCM encoder [6], the H.264 encoder searches for the best possible prediction of the encoding signal in order to reduce the prediction residue and to spend less bits in its compressed representation. The encoder may either select the prediction mode for each block that minimizes the difference between the predicted block and the block to be encoded, or to use rate-distortion (RD) optimization [7].

The search for the best possible prediction is one of the most time consuming stages of a software-based video encoder [8]. When encoding video sequences in a low-latency communications scenario, e.g. video conferencing, the time spent compressing the signal is an issue and real-time coding is challenging. As encoders take most of time doing predictions, reducing the complexity of this stage seems to be the natural way to reduce the overall complexity.

There are many works aimed at reducing AVC complexity. Some explore prediction techniques for reducing computations with small RD penalties [9, 10, 11, 12]. A recent work provides a substantive H.264/AVC complexity reduction [13], however much of the complexity scaling will not be perceived if the framework is already implemented using faster algorithms, high-performance libraries and platform dependent resources [14, 15]. Complexity scalability was also brought to a high-performance encoder [16].

In order to perform complexity-scalable rate-constrained real-time software-based video encoding, we present some modifications to non-normative aspects of the H.264/AVC prediction stage and we evaluate them using an open source high performance encoder: x264 [17].

2. ASPECTS OF X264 AVC IMPLEMENTATION

x264 is an open source H.264/AVC standard compliant implementation whose advanced computational performance is due to the use of assembly-optimized routines for the most complexity intensive operations [17]. Besides the use of highly efficient primitives in its implementation, x264 also explores a sort of early stop methodologies when performing the rate-distortion optimization tests in such a way that a 50 times speedup can be readily perceived when compared to the reference encoder[1], without penalizing RD performance.

The innovative aspect in this work is to involve the rate control mechanism in a rate-distortion-complexity (RDC) optimization framework. It is a network-aware encoder suitable to work on mobile devices, where the power constraint has to be properly addressed. Basically, rate control allows for selection of encoding parameters to maximize quality under rate and decoder video buffer constraints. The rate control in x264 is based in part upon libavcodec's implementation[2], which is mostly empirical. In this work, an average bitrate (ABR) method was chosen, which is an one-pass scheme that produces near-constant quality for a given stream [17]. As a one-pass method, the ABR method cannot exactly achieve the target bitrate because the rate control has to be done without the knowledge of future frames.

[1] "JM," Available: http://iphome.hhi.de/suehring/tml
[2] "ffmpeg," Available: http://ffmpeg.org

In order to scale the amount of complexity used to encode a particular video sequence, we choose to modify the compression process prediction stage, one of the most computational intensive steps when encoding digital video [8], as illustrated in Table 1.

Table 1. x264 relative computational complexity for “Mobile” CIF (352×288-pixel frame) and “Mobcal” 720p (1280×720-pixel frame) sequences.

	Resolution	
Coding Stage	**CIF**	**720p**
Predictions	91.24%	90.42%
Encoding	6.07%	6.13%
Other Stages	2.69%	3.45%
Total	100.0%	

Regarding the complexity settings, the x264 codec presents three parameters of interest: subpixel motion estimation and mode decision, trellis RD quantization and the available partition set. The fisrt parameter controls the amount of refinement in the motion estimation process and the use of optimized mode decision. The use of trellis RD quantization is controlled by the **- - trellis** parameter, which disables and enables the use of trellis on the final encode of a macroblock or on all mode decisions. The available partition set can be assessed through the **- -partitions** parameters.

Those parameters impact on the prediction stage performance and allow for some complexity scalability. The present work explores adapting these and other complexity related encoder parameters to more finely constrain the encoder complexity.

3. THE APPROACH

Our framework for complexity-constrained real-time video coding consists in controlling the amount of complexity spent while encoding a video sequence by adjusting the encoder in such a way that the RD penalties, compared to a full-featured case, remain as low as possible. Here, the computational complexity is measured as the time spent to encode a particular video sequence on a given system setup.

The approach is to extend an RD-optimization [18] strategy, adding the C dimension (which stands for complexity) to the regular 2-D problem of optimizing a particular codec to spend the smallest bitrate (R) necessary to represent a encoded video signal at a particular distortion (D). So, for each particular encoder setup (see 2), we compute the total bitrate (R), the average PSNR (D) and the ratio between the time spent to encode a training sequence and the the time spent by the full-featured encoder to do the same job (C).

The RDC points are used to populate an initial search space from which we look for the points that lie on its lower convex hull, as illustrated in Fig. 1 for constant rates. After finding the setups that belong to the Pareto front, we build a lookup table from the performance numbers in order to provide optimal starting RDC points. Any intermediate complexity point not found in the table can be easily achieved by interpolation in such a way that the global complexity is very close to the required complexity “budget”.

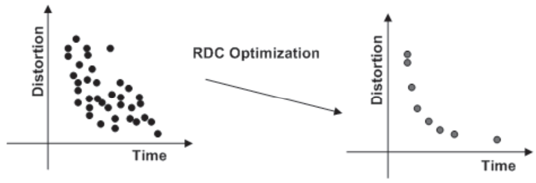

Fig. 1. Finding the set of points that compose the Pareto Front. For the illustration, rate was kept constant.

As a result of the optimization process, it is provided an extra parameter to the encoding process, **QPC**, which stands for Quantization Parameter for Complexity and tunes the encoder complexity according to the user demand by jointly adjusting the extra parameters described in Section 2. The values QPC can assume values between $(15, 100)$ and each value represents the percentage of complexity used to encode the video sequence. The lower bound for the encoding process was arbitrated as 15% of overall complexity, which was empirically found in oder to not penalize too much the encoded sequence quality.

Once found the optimal points, an extra step is to calibrate the system such that the complexity points have a corresponding encoding speed, measured as the amount of frames per seconds (**fps**) the encoder can compress. Once calibrated, the system is capable to optimally encode video (in term of RD) and constrained to a certain C, which is demanded in terms of **fps** by the user. A controller [19] is attached to the original encoder to make sure that the compression does not violate the alloted time budget (Fig. 2). Note that the input frame-skipping approach is not used in this work.

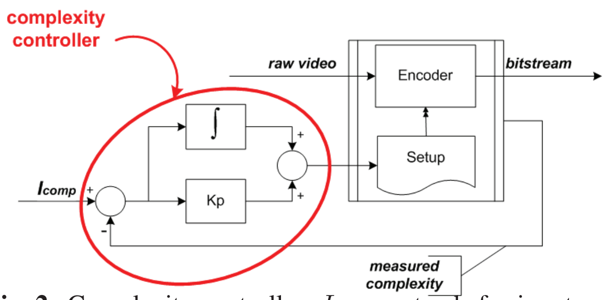

Fig. 2. Complexity controller. $Icomp$ stands for input complexity.

4. RESULTS

The proposed method was implemented in the x264 encoder, using hexagonal motion estimation, two reference frames, no B-frame and single thread encoding. Complexity was taken as the ratio of the net encoding time for the reduced complexity setup to the full-featured coder net encoding time. Initially, we evaluated the codec performance by coding CIF and 720p training sets composed by standard sequences. We tested different encoding parameters like: target bitrate, rate-distortion optimization analysis (different efficiency techniques) and subsets of macroblock partitions. The CIF tests were performed on a two-Dual-Core AMD Opteron(tm) Processor platform meanwhile the 720p ones were executed on a 2.80 GHz Core i7(tm) Processor.

For each parameter set choice, we compared its performance with the full-featured case (100%-complexity, which corresponds to approximately 9.0fps for CIF and 7.0fps for 720p sequences) by evaluating average bitrate and PSNR differences between two RD curves, as described in [20]. Then, we found the optimal RDC points which belong to the Pareto front. The general behaviour suggests that, as we reduce the complexity used to encode a video sequence, the performance penalties increase.

After training, the encoder was RDC-optimized and Figs. 3 and 4 present perfomance drop curves for test sequences for the range of complexity with starts around 20% of complexity reduction. Table 1 shows that the complexity of the remaining codecs module is around 10% of the total complexity. This lower bound is achieved only if the whole prediction process is removed from the encoder. Therefore, some complexity at the prediction module is still expected. An important observation is that the optimal performance operation point for the some sequences is not achieved at full computational complexity. This is mainly due to the fact that constraining the encoding process to certain syntax elements can better explore RD costs through more efficient entropy coding. It also yields significant encoding time reduction.

The main result is an encoding-speed controlable framework which allows the user to choose the desired encoding frame rate in **fps**. Figs. 5 and 6[3] present the RD-curves of different encoding frame rate. As expected, the RD-perfomance tends to be penalized as the encoding speed is raised. However, the curves are collocated; the worst case is represented by 60-fps (CIF) and 40-fps (720p) RD-curves, nevertheless the average PSNR penalty remains below 1dB for both resolutions. The figure also shows a flat complexity profile delivered by the framework and the controller bitrate independency.

5. CONCLUSIONS

We proposed a fully-compliant complexity-optimized framework for a H.264/AVC software implementation that allows for real-time coding. Rather than using all prediction tools provided by implementation, we can optimally choose a subset of them constrained by the amount of available complexity. Our main contribution is the insertion of an encoding parameter capable of controlling the encoding complexity and the possibility to select the desired encoding speed. Our tests have shown that the RD performance is only moderately affected by the framework, which does not make use of frame-skipping to comply to the requested encoding speed. Nevertheless, we provide significant encoding complexity reduction.

[3] Seq17 is a scene where, initially, there is a female speaker on a table and a male speaker joins in the middle of the sequence.

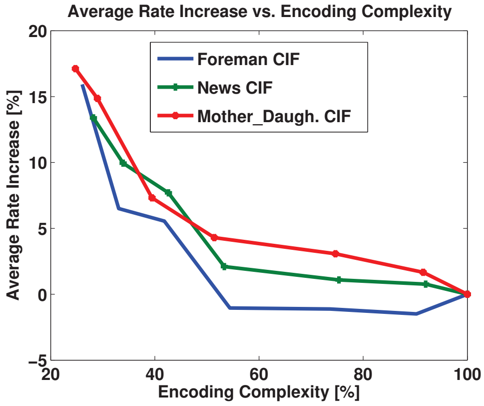

Fig. 3. Average rate increase vs. complexity savings for CIF video sequence.

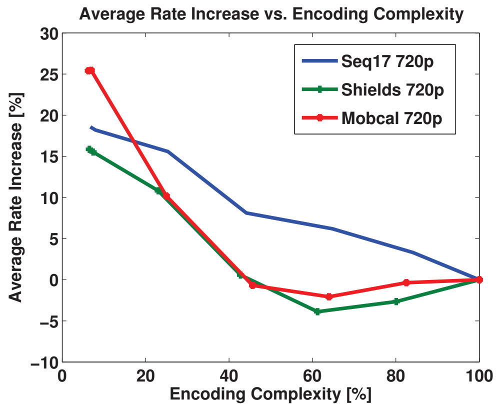

Fig. 4. Average rate increase vs. complexity savings for 720p video sequence.

In addition to the fact that out framework can be readily used to build PC-based video encoder appliances without the need of changing decoder implementation, our contribution can benefit from the availability of powerful computers for designing PC-based appliances. Also, multi-thread and multicore programming can readily fold the complexity numbers. We plan to further work on making an encoder aware of environmental and commutications conditions, capable of adjusting itself to meet channel, quality and energy constraints.

Acknowledgment

We would like to thanks Prof. Dr. Eduardo Barros da Silva and Prof. Dr. Jose Fernando Leite de Oliveira from Universidade Federal do Rio de Janeiro for providing some HD sequences.

6. REFERENCES

[1] JVT of ISO/IEC MPEG and ITU-T VCEG, "Advanced video coding for generic audiovisual services," Tech. Rep. 14496-10:2005, March 2005.

[2] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons Ltd, 2003.

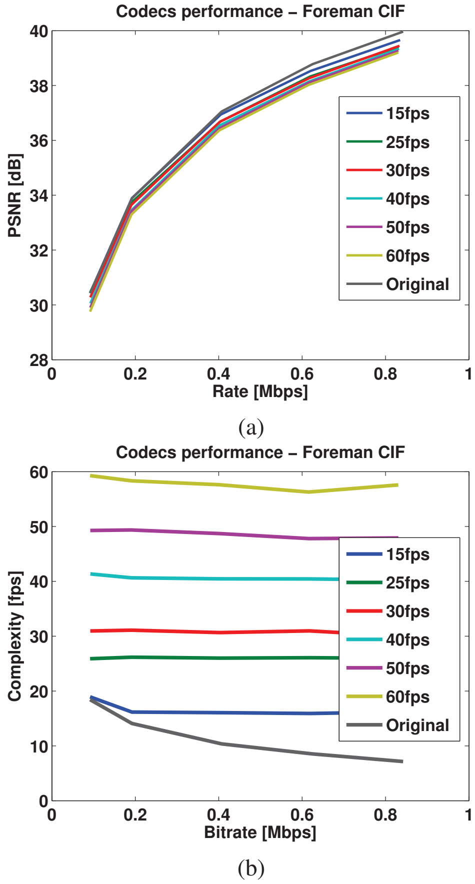

Fig. 5. Performance comparison between a full-featured codec and its encoding speed controlable version for "Foreman" sequence: (a) RD-performance and (b) Complexity profile. Original stands for the full-featured encoder.

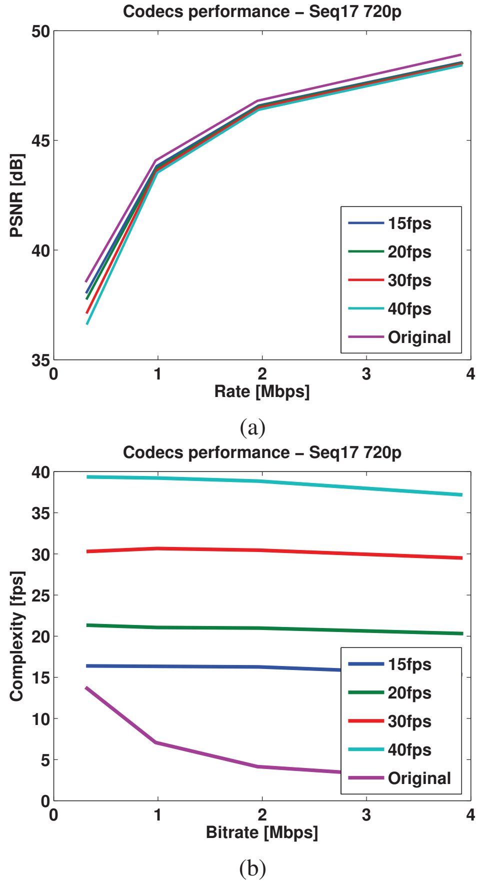

Fig. 6. Performance comparison between a full-featured codec and its encoding speed controlable version for "Seq17" sequence: (a) RD-performance and (b) Complexity profile. Original stands for the full-featured encoder.

[3] T. Wiegand, G. J. Sullivan, G. Bjontergaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[4] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions," *Proc. SPIE Conference on Applications of Digital Image Processing XXVII*, Aug. 2004.

[5] R. L. de Queiroz, R. S. Ortis, A. Zaghetto, and T. A. Fonseca, "Fringe benefits of the H.264/AVC," in *Proc. International Telecommunication Symposium*, 2006, pp. 208–212.

[6] K. Sayood, *Introduction to Data Compression*, Morgan Kuffmann Publishers, 2000.

[7] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, pp. 7–28, Jan-Mar 2004.

[8] Y.-Y. Huang, B.-Y. Hsieh, S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 4, pp. 507–522, Apr. 2006.

[9] Z. Chen, P. Zhou, and Y. He, "Hybrid unsymmetrical-cross multi-hexagon-grid search strategy for integer pel motion estimation in H.264," *Proc. Picture Coding Symposium*, Apr. 2003.

[10] B.G. Kim, S.-K. Song, and C.-S. Cho, "Efficient inter-mode decision based on contextual prediction for the P-slice in H.264/AVC video coding," *Proc. IEEE International Conference on Image Processing*, pp. 1333–1336, September 2006.

[11] C. S. Kannangara, Y. Zhao, I. E. Richardson, and M. Bystrom, "Complexity control of H.264 based on a bayesian framework," *Proc. Picture Coding Symposium*, November 2007.

[12] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 645–658, May 2005.

[13] M. Moecke and R. Seara, "Sorting rates in video encoding process for complexity reduction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, pp. 88–101, 2010.

[14] Kyu Park, Nitin Singhal, Man Hee Lee, , and Sungdae Cho, "Efficient design and implementation of visual computing algorithms on the GPU," *Proc. IEEE Intl. Conf. on Image Processing*, November 2009.

[15] Intel, "Intel Integrated Performance Primitives," *http://software.intel.com/en-us/intel-ipp/*.

[16] M.C. Chien, J.Y. Huang, and P.C. Chang, "Complexity control for H.264 video encoding over power-scalable embedded systems," in *Proc. 13th IEEE Internatinal Symposium on Consumer Electronics, ISCE2009*, 2009, pp. 221–224.

[17] L. Merritt and R. Vanam, "Improved rate control and motion estimation for H. 264 encoder," in *IEEE International Conference on Image Processing, 2007. ICIP 2007*, 2007, vol. 5.

[18] Gary J. Sullivan and Thomas Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, November 1998.

[19] Katsuhiko Ogata, *Modern Control Engineering*, Prentice Hall, 5th edition, 2009.

[20] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33*, April 2001.