Complexity Reduction Techniques for the Compression of High-Definition Video

Tiago A. da Fonseca and Ricardo L. de Queiroz

Abstract— **The state-of-the-art video compression standard, the hybrid predictive-transform H.264/AVC codec, has lead to substantial performance improvement compared to other existing standards. Performing predictions is a computational demanding task and optimizing this stage may result in substantial encoding speedup. In this paper, we propose more efficient approaches to implement the H.264/AVC prediction stage. The first idea is to use original data rather than reconstructed ones to perform prediction tests before choosing the best mode. The residue, however, is evaluated using previously decoded data in order to avoid drifting. The second approach is to employ a subset of dominant prediction modes instead of testing all modes recommended by the H.264/AVC standard. The subset is updated frame-by-frame using macroblock sampling. Results for high definition sequences show that the quality loss is negligible allowing us not only to parallelize the inter-prediction stage but also to reduce total complexity.**

Index Terms— **H.264/AVC, inter-frame prediction, reduced complexity.**

I. Introduction

HISTORICALLY, processor manufacturers have responded to the demand for more processing power primarily with faster processor speeds. However, higher clock speeds imply in higher power consumption and heat. For that reason, manufacturers have been moving its strategy from pure clock-oriented projects to embrace multi-core architectures.

Image and video processing can be considered as driven forces that motivate this computational power pursuit. Therefore, it is not surprisingly to observe that the *state of art* video compression standard, H.264/AVC [1], is a computation-hungry application. The H.264/AVC coder has been well described in the literature [2]-[3].

When encoding high definition sequences, complexity is an issue and real-time video coding is challenging. As the computational complexity of H.264/AVC is mainly concentrated in the prediction stage, making it more efficient seems to be a key to allow for real-time coding. The present work suggests new strategies in this direction. The first idea proposes to parallelize the prediction module to allow for real-time coding exploring the advent of multi-core architectures. Another idea is to suppress least frequent prediction modes in order to save complexity.

This paper is organized as follows. Section II gives an overview of the H.264/AVC macroblock prediction and analyses prediction complexity. A method to reduce the computational complexity is presented in Section III. The experimental results are shown and discussed in Section IV, while the conclusions are finally presented in Section V.

II. Macroblock Prediction in H.264/AVC

H.264/AVC is a hybrid video codec, i.e. along with a transform module, it has a prediction module, a differential stage and a feedback loop [2]. The prediction stage uses previously reconstructed samples as input to the prediction model. This avoids mismatches between encoder and decoder data, allowing for synchronous decoding. However, H.264/AVC has a rather complex prediction stage composed by a set of prediction models.

In Fig. 1, the H.264/AVC encoder block diagram is shown and the prediction stage is highlighted. Note that the coder is divided into temporal (Inter) and spatial (Intra) models. “Inter” prediction generates a prediction macroblock from one or more previously encoded video frames using block-based motion estimation and compensation. This model is responsible for almost 90% of the complexity of an H.264/AVC baseline encoder [4]. Important advances from earlier video standards include the support for a range of block sizes (16×16 and down, as in Fig. 2), and refined motion vectors (quarter-sample resolution for the luminance component). In “Intra” prediction, a prediction block is formed based on planar extrapolation of previously encoded and reconstructed neighbouring pixels. The prediction is subtracted from the current block, prior to encoding. The 4×4- and 8×8-pixel blocks allow for a total of nine optional prediction modes for luminance, while the 16×16-pixel blocks allow for only four modes as illustrated in Fig. 3. The encoder typically selects the prediction mode for each block that minimizes the difference between the predicted block and the block to be encoded.

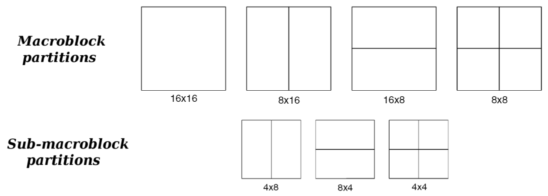


Fig. 2. Macroblock and submacroblock partitions for motion compensation in Inter Prediction.

A prediction for the current macroblock is created from image samples that have already been encoded either in the same frame or in a previously encoded one. This prediction is subtracted from the current macroblock and the residue is compressed and transmitted, along with information required

Manuscript received September 19, 2008; revised November 4, 2008. The authors are with Department of Electrical Engineering, Universidade de Brasilia, e-mail tiago@image.unb.br and queiroz@ieee.org. This work was supported by HP Brasil.

ISSN 1980-6604/$20.00 © 2009 SBrT/IEEE

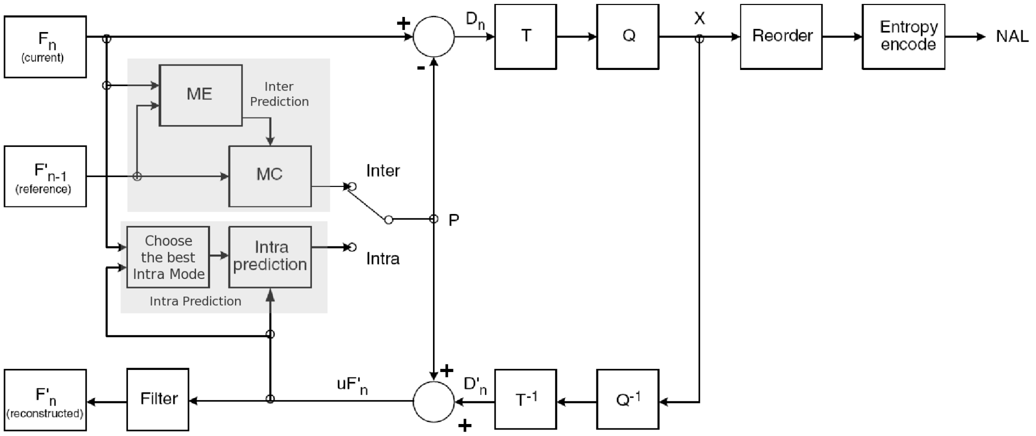
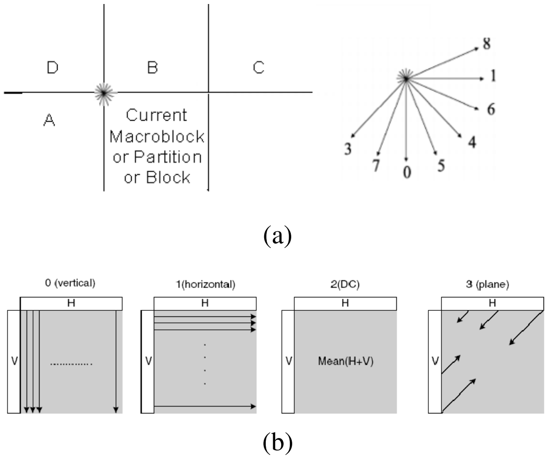


Fig. 1. H.264/AVC encoder block diagram indicating the prediction stage.



(a)

(b)

Fig. 3. Intra prediction modes and their respective planar extrapolation directions for (a) 4×4/8×8 blocks and (b) 16×16 blocks. In (a), mode 2 is DC prediction.

by the decoder in order to repeat the prediction process (motion vectors, prediction modes, etc.). The decoder creates an identical prediction as the encoder, and adds it to the decoded residual block. The encoder bases its prediction on encoded and decoded image samples (rather than on original video frame samples) in order to ensure that the encoder and decoder predictions are identical.

The H.264/AVC prediction stage is built upon a myriad of tests applied to choose the best prediction mode in a rate-distortion (RD) sense. It is intuitive that encoding complexity reduction can be achieved through simplifying the prediction module, particularly the motion estimation step.

Sub-optimal fast motion estimation techniques were proposed [5], [6] and incorporated in the H.264/AVC reference software[1]. In exploring the variety of macroblock partitions available in H.264/AVC, there are works [7], [8] that apply motion estimation only for the most probable partition. Intra-prediction tests can also be reduced by means of selection of the most probable best mode according to heuristics [9], [10]. Another approach is to generalize the rate-distortion analysis to add a complexity optimization variable. This concept is well suited to the emerging field of wireless digital video communications, where energy and delay constraints are stringent [11],[12].

In order to profile the H.264/AVC (*High Profile*) encoder we used `gprof`[2] and JM12.3 to encode a high definition video sequence (Pedestrian Area), with rate-distortion optimization turned on, four-frame reference buffer and *fast full-search* motion estimation [1]. Results are presented in Tables I and II.

TABLE I
RELATIVE COMPUTATIONAL COMPLEXITY FOR HD PEDESTRIAN AREA INTRA-FRAME ONLY, IN JM12.3 H.264/AVC *High Profile* CODING.

Coding Stage	Percent [%]
4×4 Intra Prediction	25,2
8×8 Intra Prediction	20,9
16×16 Intra Prediction	6,1
Other Stages	47,8
Total	100,0

TABLE II
RELATIVE COMPUTATIONAL COMPLEXITY FOR HD PEDESTRIAN AREA IN JM12.3 H.264/AVC *High Profile* CODING AND VARIOUS MOTION ESTIMATION SEARCH WINDOW SIZES.

	Window Size (pixels)				
Coding Stage	8	16	32	64	128
Motion Estimation	65,8%	78,5%	90,8%	97,1%	99,1%
4×4 Intra Prediction	6,1%	3,8%	1,6%	0,5%	0,1%
8×8 Intra Prediction	4,9%	3,2%	1,3%	0,4%	0,1%
16×16 Intra Prediction	1,4%	1,0%	0,4%	0,1%	0,0%
Other Stages	21,8%	13,5%	5,9%	1,9%	0,7%
Total			100,0%		

Table I indicates H.264/AVC Intra-prediction modes complexity contributions. We observe that the prediction complexity for 4×4 and 8×8-pixel blocks is greater than 16×16 ones.

[1] JM Available: http://iphome.hhi.de/suehring/tml/

[2] http://www.gnu.org/software/binutils/

This is due to the larger number (nine) of prediction modes available for 4×4 and 8×8 blocks.

Complexity estimates presented in [4] are extended in Table II, where different motion search window sizes were applied to H.264/AVC high definition video sequence encoding. We observe that the encoder spends great part of execution time in *Motion Estimation* due to the extensive tests required to find the best match.

III. Complexity-Efficient Coding

A. Open-loop to enable parallelism

In the HD video coding, usually only a low level of distortion is tolerated, which implies a high degree of similarity between the compressed video and its original version. So, if we employ original data in prediction tests instead of reconstructed ones, it is likely that the best prediction mode chosen using original data will be the same mode chosen when using reconstructed data. The proposed method, described in Fig. 4, consists in applying original data to choose the best prediction mode, but to use locally decoded data in the motion compensation process, in order to avoid drifting.

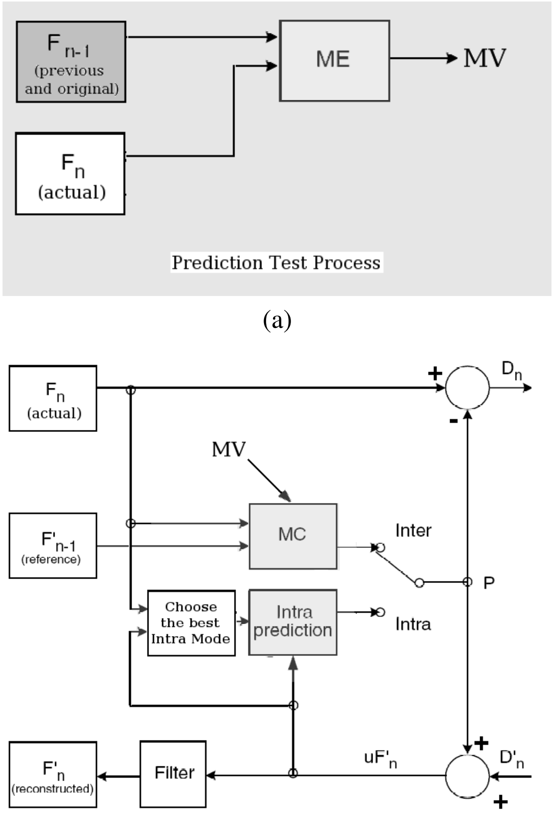
In Fig. 4, the prediction tests do not depend on reconstructed/decoded data as compared to Fig. 1. They only depend on original data. The search for the best prediction mode, the most time consuming stage of an H.264/AVC encoder, can be parallelized given that we do not need to wait for previously encoded/decoded data. Thus, the prediction loop is opened (see Fig. 4). All data/modes used to evaluate the residue are available to the decoder. One can potentially reduce encoding time by a factor of n through the parallel work of n frame prediction engines.

Some aspects of encoder implementation on PC-based platforms still have to be addressed. High-definition encoding requires the processing of a huge amount of data, which should be stored in memory and also transferred to and from the processor. Therefore, bus throughput may be a bottleneck to the whole process.

B. Prediction Mode Bias

The H.264/AVC prediction stage is rather complex due to the many tests of various prediction modes available to each macroblock. For instance, to encode P-frames in H.264/AVC High Profile, we can use the following set of Inter- and Intra-frame prediction modes:

- P16×16: motion compensated prediction for 16×16 pixels macroblocks;
- P16×8: motion compensated prediction for 16×8 pixels macroblocks;
- P8×16: motion compensated prediction for 8×16 pixels macroblocks;
- P<=8×8: motion compensated prediction for macroblocks whose size is less then or equal to 8×8 pixels;
- I16MB: intra prediction for 16×16 pixels macroblocks;
- I8MB: intra prediction for 8×8 pixels macroblocks;
- I4MB: intra prediction for 4×4 pixels macroblocks;
- SKIP: zero residue motion compensated prediction for 16×16 pixels macroblocks.



(b)

Fig. 4. Proposed parallel prediction structure (a) and the new encoder scheme for parallel prediction.

When compressing high definition 1080p video sequences (1920×1080 pixels per frame in progressive scan), we verify that the prediction modes used to encode the signals are often repetitive. The frequency profile of selected prediction modes for different sequences and resolutions, ranging from QCIF (176×144 pixels) to 1080p, is presented in Figs. 5 through 7.

We can observe that as we increase the resolution, prediction modes tend to polarize themselves around larger macroblock partitions, even though the Riverbed sequence does not strictly follow this trend. In general, some computational effort can be saved when encoding high definition video sequences by avoiding small-sized partitions in motion compensated predictions.

C. Reduced Mode Set Prediction

The previous analysis suggests that the encoder can save time if it avoids testing the less frequent prediction modes. In order to achieve complexity reduction based on frequency distribution of best prediction modes, we randomly select macroblocks to preview the frequency distribution of the next frame. Then, we select the dominant modes, i.e. the modes which correspond to 80% of the choices, according to the following algorithm:

```
Let each frame have $N$ macroblocks. For the $n$-th P- or
  B-frame
1. Randomly select a set $S$ of $N_S$ macroblocks
```

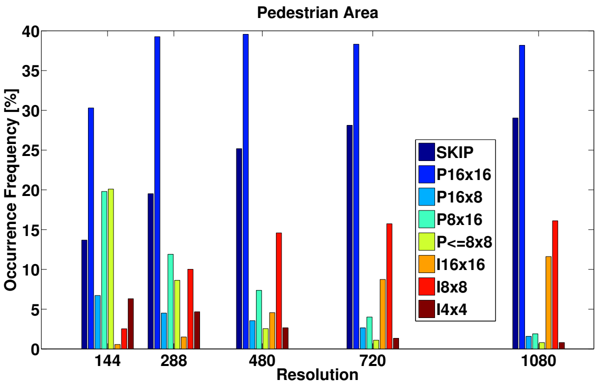


Fig. 5. Best prediction modes occurrence frequency for Pedestrian Area sequence.

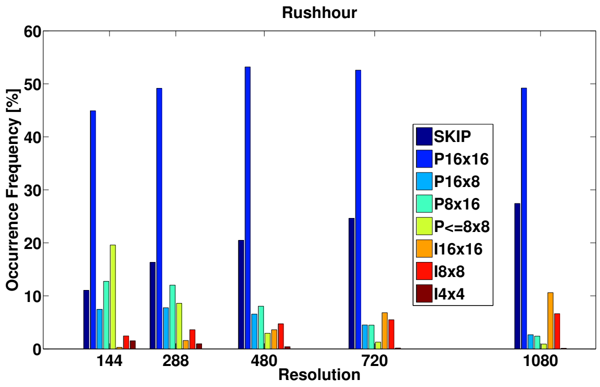


Fig. 7. Best prediction modes occurrence frequency for Rushhour sequence.

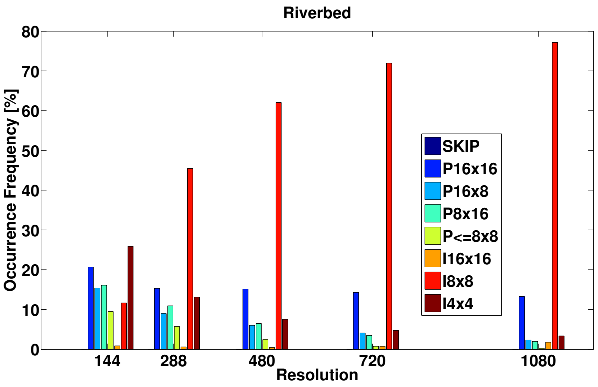


Fig. 6. Best prediction modes occurrence frequency for Riverbed sequence.

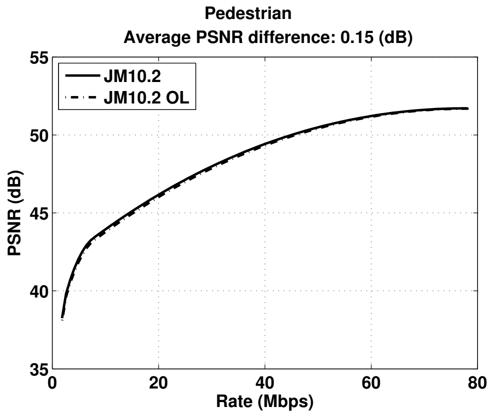


Fig. 8. Rate-distortion curves for HD sequence Pedestrian Area. OL stands for open-loop.

of the n-th frame. The remaining $N - N_S$ macroblocks form the complement set S'.

2. Test all prediction modes for macroblocks in set S, in order to pick the best mode.
3. Test only dominant modes for macroblocks in S'.
4. Set dominant mode set D as empty.
5. Rank best modes in set S.
6. Assign the most chosen modes to D until they could account to at least $0.8 \times N$.
7. Set $n \leftarrow n + 1$ and repeat.

Even though the best mode frequency distribution is not stationary, our tests have shown that this is a good approximation for adjacent frames. Errors in determining the dominant modes will be reflected in a small degradation on the encoder RD performance.

An important issue is the sampling population size, N_S, which will be used in the prediction of next frame dominant modes. The smaller N_s is, the larger the savings, but the worst the performance.

IV. Experimental Results

A. Open-Loop Prediction

The open-loop prediction method was implemented in JM10.2[3] and evaluated using the first 50 frames of standard test sequences Pedestrian Area, Sunflower, Riverbed and Rushhour. We varied the QP (quantization parameter) in the range $12 \leq QP \leq 36$. RD plots are presented in Figs. 8 through 11, comparing the methods with and without an open-loop [13]. The average PSNR differences between RD-curves are indicated. From the plots, we can observe a negligible quality loss when using original data rather than reconstructed ones in the H.264/AVC prediction step. This is due to eventual mismatches between the best prediction modes. At high rates, the RD curves tend to overlap. This is expected because the lesser the quantization, the closer the original and reconstructed blocks.

The technique was also evaluated at CIF (352×288 pixels) resolution (see Figs. 12 through 14). In this case, performance losses are more significant due to prediction mode mismatches.

The technique was also evaluated for an IPBBBPBBP... GOP configuration, where 21 frames of each HD video sequence were encoded. The results are presented in Fig. 15 to 18. In this case, performance drop appears to be very small and may be acceptable for most applications.

B. Reduced Mode Set Prediction

The statistical analyzer proposed in Sec. III-C was implemented in JM12.3 H.264/AVC reference software. For each sequence we used 20 frames. Fast full-search motion estimation was used and the results were obtained by varying the

[3] JM Available: http://iphome.hhi.de/suehring/tml/

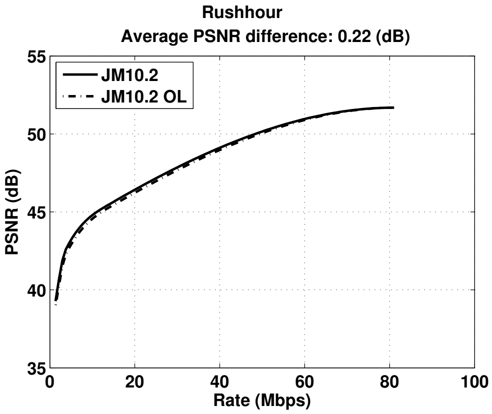


Fig. 9. Rate-distortion curves for HD sequence Rush Hour. OL stands for open-loop.

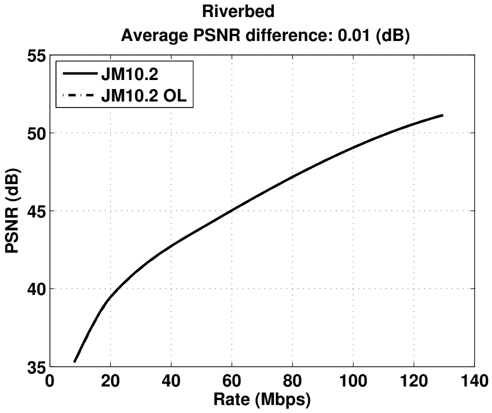


Fig. 10. Rate-distortion curves for HD sequence Riverbed. OL stands for open-loop.

QP (quantization parameter) over the range $12 \leq QP \leq 36$. Figure 19 presents RD performance curves for "Pedestrian Area" sequence, for different sampling population $\left(\frac{N_S}{N}100\%\right)$. We observe that the performance difference is very small.

A more detailed approach is to plot the average difference between the performance curves for different sampling population sizes, as show in Figs. 20 and 21. The average PSNR and bitrate differences between RD-curves were evaluated. [13]

The time savings are shown in Fig. 22, which relates complexity savings against the population size of fully-tested macroblocks, using the proposed method for different HD sequences.

Figure 22 suggests that the size of the fast-predicted macroblock population for the set of test sequences has a direct relation to the complexity savings.

There is a small performance loss when predicting only through dominant modes due to eventual mismatches. The rate loss is kept below 5% if the population size remains above 10%. Even though computational savings are relatively small, fast full-search motion estimation was already enabled. The encoder carries on 4×4-pixel block motion estimation and proceeds to larger partitions motion estimation by grouping the results (SAD/SSD) of previously stored blocks. Thus, for sequences where the intra-predicted macroblocks are more frequent, like Riverbed, the computational savings are greater due to the fact that motion estimated prediction modes are not included in the dominant set for some frames.

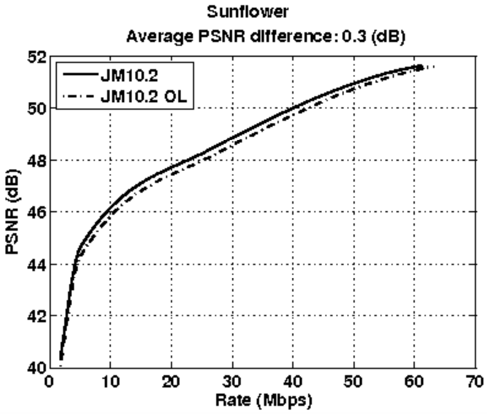


Fig. 11. Rate-distortion curves for HD sequence Sunflower. OL stands for open-loop.

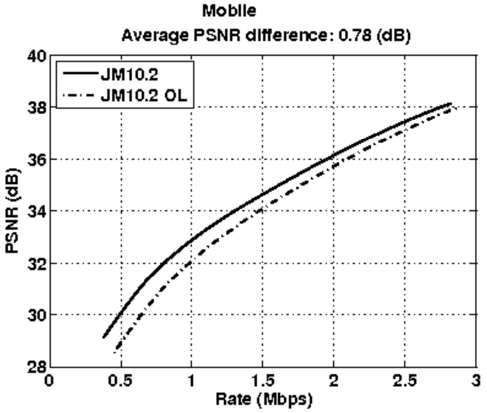


Fig. 12. Rate-distortion curves for CIF sequence Mobile. OL stands for open-loop.

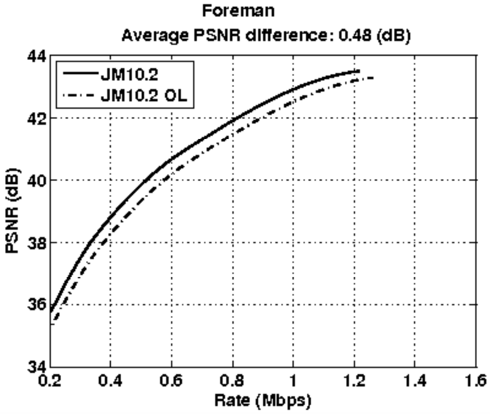


Fig. 13. Rate-distortion curves for CIF sequence Foreman. OL stands for open-loop.

We also implemented our method for UMHexS motion estimation [5] in JM13.2 H.264/AVC reference software. We computed 50 frames of each sequence and varied the population size from 1% to 50%. Results are presented in Figs. 23 through 25. The computational savings are greater than for

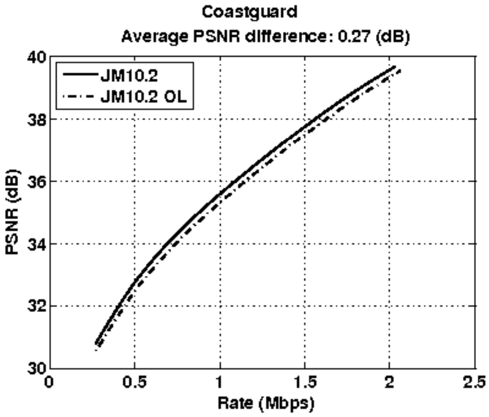


Fig. 14. Rate-distortion curves for CIF sequence Coastguard. OL stands for open-loop.

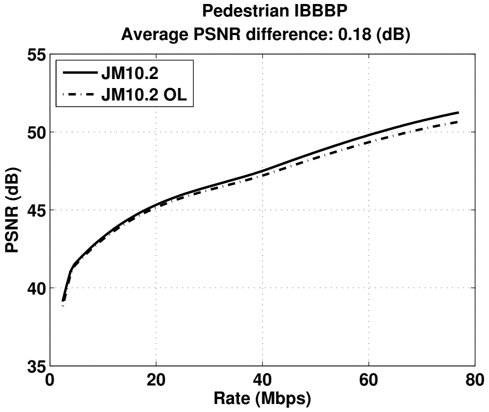


Fig. 15. Rate-distortion curves for HD sequence Pedestrian Area encoded according to IPBBBPBBP... GOP. OL stands for open-loop.

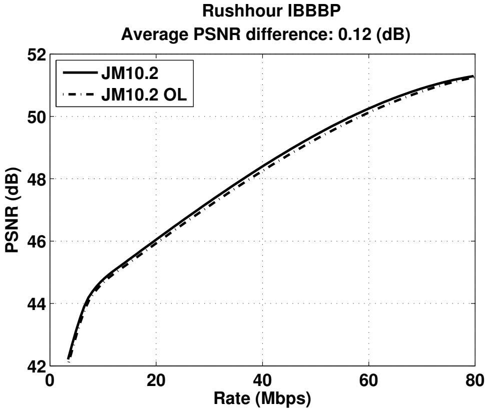


Fig. 16. Rate-distortion curves for HD sequence Rush Hour encoded according to IPBBBPBBP... GOP. OL stands for open-loop.

fast full-search case. Nevertheless, the rate and quality losses have also increased.

For "Riverbed" sequence, the complexity reduction profile is shown in Fig. 26. This sequence is very challenging because of the difficulty in getting good matches through motion estimation, incurring in a high number of intra-coded macroblocks. This characteristic was properly "tracked" by our algorithm. Complexity reduction profile plots from HD

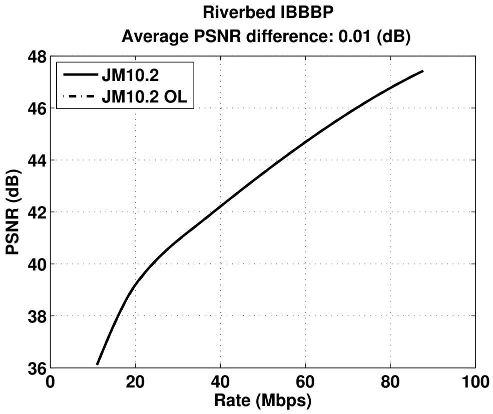


Fig. 17. Rate-distortion curves for HD sequence Riverbed encoded according to IPBBBPBBP... GOP. OL stands for open-loop.

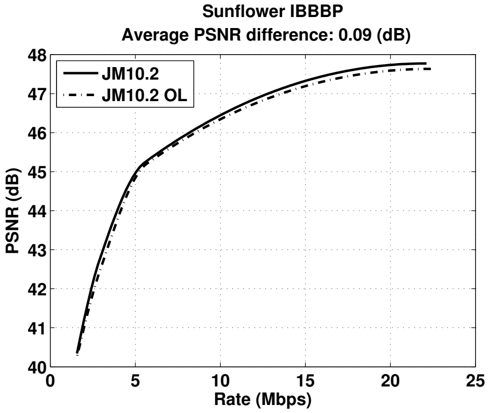


Fig. 18. Rate-distortion curves for HD sequence Sunflower encoded according to IPBBBPBBP... GOP. OL stands for open-loop.

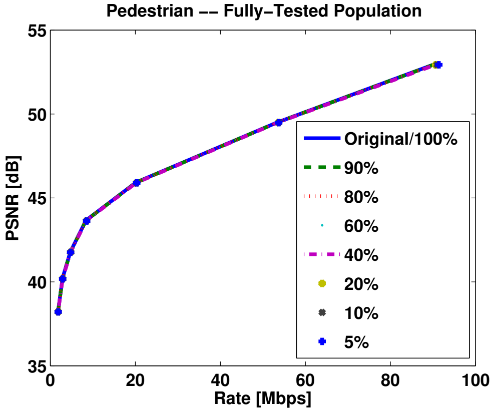


Fig. 19. Different tested population size rate-distortion curves for Pedestrian Area. The curves are essentially co-located.

sequence Sunflower are shown in Fig. 27.

A more general GOP configuration wass used to evaluate the method performance when encoding B-frames. The first 21 frames from the previous HD sequences were encoded using JM12.3 and the following presets: RD-optimization turned on, 4-frame reference buffer, UMHexS motion estimation and IPBBBPBBBP.... GOP configuration. Results are presented in Figs. 28 through 30

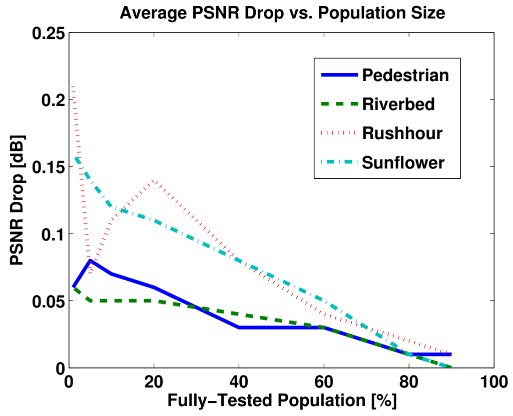


Fig. 20. Average PSNR Drop *vs.* Population size for different HD video sequences.

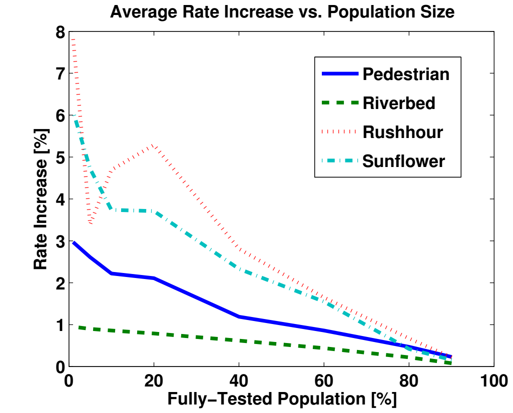


Fig. 21. Average Rate Increase *vs.* Population size for different HD video sequences.

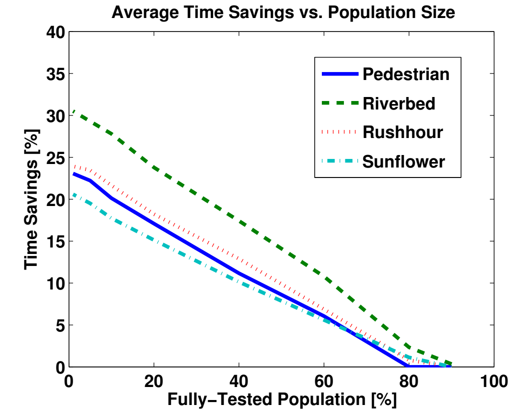


Fig. 22. Time savings *vs.* Population size for different HD video sequences.

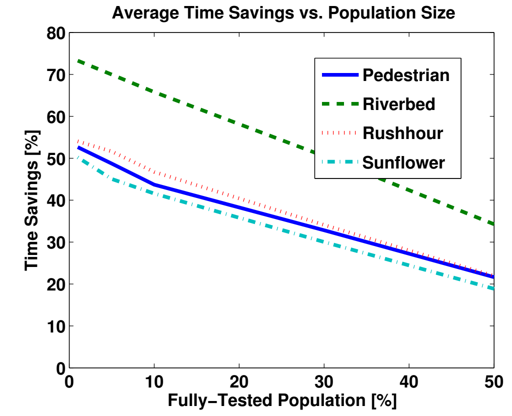


Fig. 23. Time savings *vs.* Population size for different HD video sequences using UMHexS motion estimation.

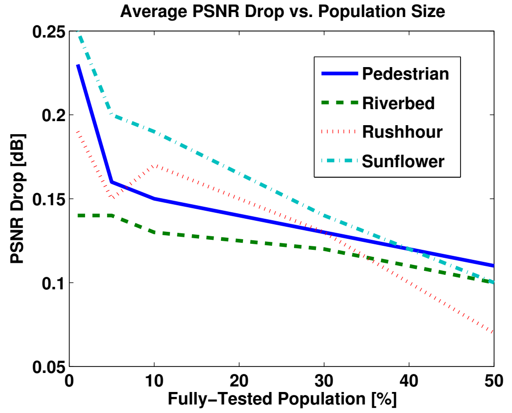


Fig. 24. Average PSNR Drop *vs.* Population size for different HD video sequences using UMHexS motion estimation.

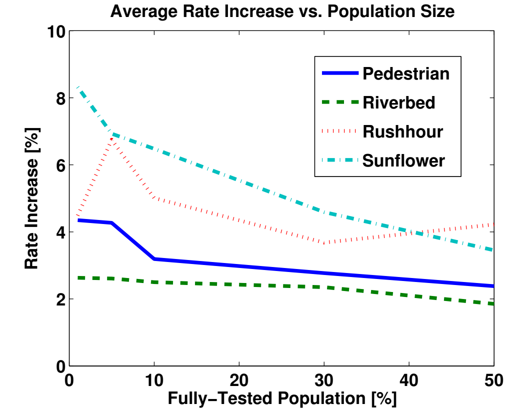


Fig. 25. Average Rate Increase *vs.* Population size for different HD video sequences using UMHexS motion estimation.

From Figs. 28 to 30 we observe that the computational savings, besides inferior to *UMHexS* results, remain greater than for *fast full-search* case. Nevertheless, the rate and quality losses have also increased. Again, the algorithm was capable of tracking some specific features from the test sequences: “Riverbed” suffered lesser penalties and achieved the best complexity savings because the algorithm could avoid tests of motion compensated modes due to their lower statistic occurrence.

The performance of the proposed technique can still be improved because the authors believe that separating the dominant modes obtained from P-frames from the ones obtained from B-frames allows for a better match of the next dominant modes. In the actual implementation, the motion compensated frames use previous frames statistics to decide which are the dominant modes; we suggest that P-frames statistics may be used to predict dominant modes in the next P-frame meanwhile

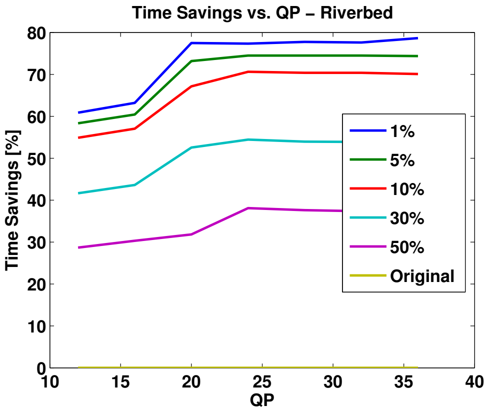


Fig. 26. Complexity reduction profile for Riverbed using UMHexS motion estimation.

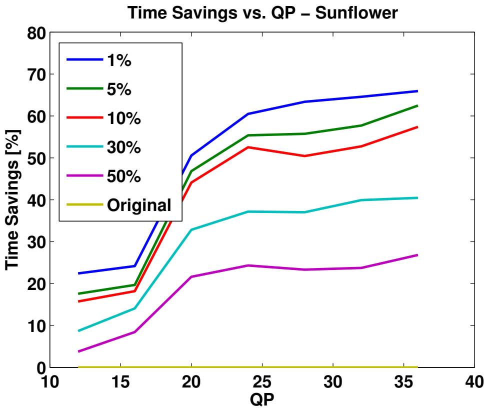


Fig. 27. Complexity reduction profile for Sunflower using UMHexS motion estimation.

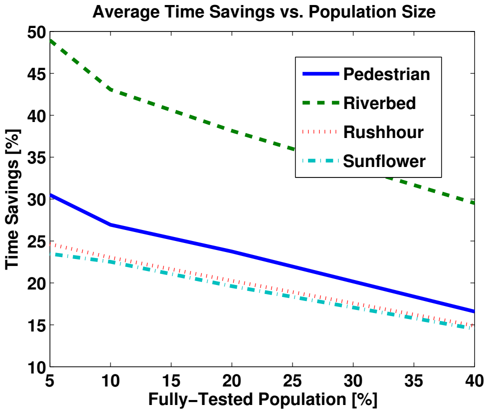


Fig. 28. Time savings *vs.* Population size for different HD video sequences using IPBBBPBBP... GOP.

B-frames statistics may be used to predict dominant modes in the next B-frame.

V. Conclusions

We propose a method to search for the best prediction mode in H.264/AVC for high-definition sequences. Rather than using previously decoded macroblocks, we propose to use the original macroblocks. In other words, we open the prediction loop. While the original image data is used for the prediction mode decision, the residue is formed using locally decoded data. Hence, drifting is avoided. In our tests, the performance loss is negligible in most cases. Its main advantage is that it allows for the parallelization of the implementation since all prediction modes and motion vectors can be tested simultaneously. This may enable real-time H.264/AVC compression of HD material using massively-parallel computing systems that do not share memory buses.

Another contribution is a reduced-complexity method to carry the prediction mode tests in H.264/AVC for high-definition sequences. Instead of testing all available prediction modes, we search for a “dominant” mode subset. Our tests have shown that the RD performance is barely affected by the prediction mode test skipping, while achieving significant complexity reduction. The method does not require a new decoder implementation because only non-normative codec aspects are modified.

As a future research, we plan to implement a complexity-controlled H.264/AVC encoder based on a test-skipping strategy. We also plan to benchmark video compression on massively-parallel computing systems.

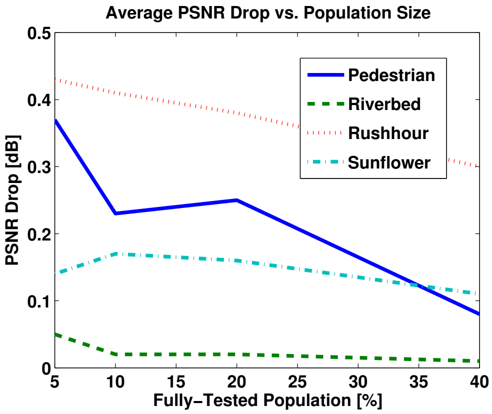


Fig. 29. Average PSNR Drop *vs.* Population size for different HD video sequences using IPBBBPBBP... GOP.

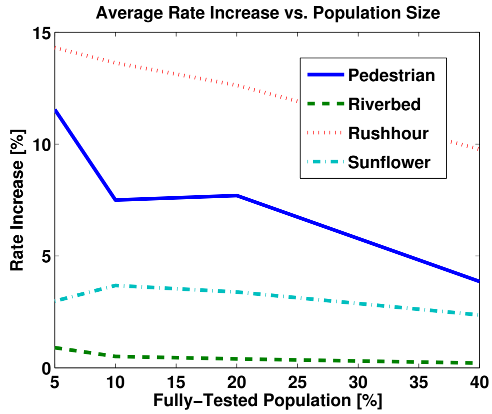


Fig. 30. Average Rate Increase *vs.* Population size for different HD video sequences using IPBBBPBBP... GOP.

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.

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

[4] 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, April 2006.

[5] Z. Chen, P. Zhou, and Y. He, "Hybrid Unsymmetrical-cross Multi-Hexagon-grid Search Strategy for Integer Pel Motion Estimation in H.264," *Picture Coding Symposium*, April 2003.

[6] H.-Y. C. Tourapis, A. M. Tourapis, and P. Topiwala, "Fast Motion Estimation within the H.264 Codec," *Proc. of International Conference on Multimedia and Expo. ICME*, vol. 3, pp. 517–520, July 2003.

[7] T.-Y. Kuo and C.-H. Chan, "Fast Variable Block Size Motion Estimation for H.264 Using Likelihood and Correlation of Motion Field," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1185–1195, October 2006.

[8] B. 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," *IEEE International Conference on Image Processing*, pp. 1333–1336, September 2006.

[9] B. La, M. Eom, and Y. Choe, "Fast Mode Decision for Intra Prediction in H.264/AVC Encoder," *IEEE International Conference on Image Processing*, vol. V, pp. 321–324, September 2007.

[10] C. Hwang, S. Zhuang, and S.-H. Lai, "Efficient Intra Mode Selection Using Image Structure Tensor for H.264/AVC," *IEEE International Conference on Image Processing*, vol. V, pp. 289–292, September 2007.

[11] 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.

[12] E. Akyol, D. Mukherjee, and Y. Liu, "Complexity Control for Real-Time Video Coding," *IEEE International Conference on Image Processing*, vol. I, pp. 77–80, September 2007.

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

Ricardo L. de Queiroz received the Engineer degree from Universidade de Brasilia, Brazil, in 1987, the M.Sc. degree from Universidade Estadual de Campinas, Brazil, in 1990, and the Ph.D. degree from The University of Texas at Arlington , in 1994, all in Electrical Engineering.

In 1990-1991, he was with the DSP research group at Universidade de Brasilia, as a research associate. He joined Xerox Corp. in 1994, where he was a member of the research staff until 2002. In 2000-2001 he was also an Adjunct Faculty at the Rochester Institute of Technology. He joined the Electrical Engineering Department at Universidade de Brasilia in 2003. He is now Full Professor at the Computer Science Department at Universidade de Brasilia.

Dr. de Queiroz has published over a 130 articles in Journals and conferences and contributed chapters to books as well. He also holds over 45 issued patents. He is a member of the IEEE Signal Processing Society's Image, Video and Multidimensional Signal Processing (IVMSP) Technical Committee and of the Multimedia Signal Processing (MMSP) Technical Committee. He is an associate editor for the IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology and a past editor for IEEE Signal Processing Letters. Dr. de Queiroz has been actively involved with the Rochester chapter of the IEEE Signal Processing Society , where he served as Chair and organized the Western New York Image Processing Workshop since its inception until 2001. He is the General Chair of ISCAS'2011 and of MMSP'2009. He was also part of the organizing committee of ICIP'2002. His research interests include image and video compression, multirate signal processing, and color imaging. Dr. de Queiroz is a Senior Member of IEEE, a member of the Brazilian Telecommunications Society and of the Brazilian Society of Television Engineers

Tiago A. da Fonseca received the B.Eng. and the M.S. degrees in electrical engineering from the Universidade de Brasilia, Brasilia, Brazil, in 2006 and 2008, respectively. He is currently pursuing the D.Sc. degree at the Department of Electrical Engineering, University of Brasilia.

His main research interests are in image processing, video processing and scalable video coding.