ON OVERRIDING H.264/AVC B-SLICE PREDICTED RESIDUE CODING

Nuno M. M. Rodrigues[1,2], *Eduardo A. B. da Silva*[3], *Murilo B. de Carvalho*[4],
Sérgio M. M. de Faria[1,2], *Vitor M. M. da Silva*[1,5]

[1]Instituto de Telecomunicações, Portugal; [2]Instituto Politécnico Leiria - ESTG, Portugal;
[3]PEE/COPPE/DEL/Poli, Univ. Fed. Rio de Janeiro, Brazil; [4]TET/CTC, Univ. Fed. Fluminense, Brazil;
[5]DEEC, Universidade de Coimbra, Portugal.

e-mails: nuno.rodrigues@co.it.pt, eduardo@lps.ufrj.br, murilo@telecom.uff.br, sergio.faria@co.it.pt, vitor.silva@co.it.pt

ABSTRACT

The use of motion compensation (both forward and bidirectional) allied with transform-quantisation based coding of the residual predicted error has been ubiquitous in video coding standards. The most recent standard, H.264/AVC, maintains this paradigm, introducing even more efficient tools to determine and compress predicted residue information.

In this paper we investigate the effects of eliminating the encoding of bidirectional prediction residue in H.264/AVC. Extensive experimental tests show gains of up to 50% of the bit rate for the same PSNR value, for B slices. An analysis of this procedure also reveals an interesting relation between H.264/AVC motion estimation and dictionary-based methods, namely vector quantisation and Lempel-Ziv encoders.

***Index Terms*—** H.264/AVC, Video coding, Vector Quantisation, Lempel-Ziv

1. INTRODUCTION

Several decades of video coding standards have confirmed the hybrid model as the preferred architecture for video encoding algorithms. In these methods, a motion compensation (MC) stage reduces the temporal redundancy of the signal. The motion compensated residue is then compressed using traditional transform-quantisation-entropy coding methods, that efficiently explore the data's spatial correlation.

This general architecture was maintained in the successful H.264/AVC [1] standard, that achieves more than twofold gains over its predecessors. A relevant fact is that these performance gains are not the result of a change of coding paradigm; it comes mainly from the exploitation of a richer set of tools for each of the encoders' modules, resulting in a more complex, but highly efficient method [2].

Arguably, the prediction process of H.264/AVC represents a very significant evolution in relation to the previous encoders. For Intra macroblocks (MB) a set of directional spatial prediction modes is used to estimate the image data. For Inter frame prediction, new adaptive block size techniques were introduced, allowing for the use of small block sizes; motion vectors (MV) can be represented up to quarter-pixel accuracy and may point beyond picture boundaries; several frames may be used as reference for MC, that may also use a weighted average of prediction frames. The use of a rate-distortion (RD) optimisation algorithm enables the choice of the most favourable method for each block. Important gains are also achieved by new entropy coding methods and enhanced techniques for coding the residual error, namely a new integer and invertible transform, the use of adaptive (and smaller) block sizes and a hierarchical block transform.

For bidirectional (*B*) slices, the combination of these techniques has impressive results in the compression of video data. In fact, the use of motion compensated prediction using partitions with adaptive dimensions, of both past and future reference frames, yields a residual error information that can be effectively compressed.

Based on these observations, in this paper we investigate the performance of H.264/AVC video encoder when no bidirectional MC residual data is encoded, as described in section 2. Section 3 describes a formal analysis relating the resulting method with dictionary-based encoding schemes, like vector quantisation (VQ) or Lempel-Ziv (LZ) algorithms. Experimental tests performed using a set of well know test sequences demonstrate that, for *B* slices, this process achieves important gains over the original H.264 encoder. These results are presented and discussed in section 4. Some conclusions are drawn in section 5.

2. OVERRIDING MOTION-COMPENSATED RESIDUAL CODING

H.264/AVC allows for seven different segmentation modes of the motion compensated block, organised in two levels: luma blocks with $16{\times}16$, $16{\times}8$, $8{\times}16$ and $8{\times}8$ can be used. For $8{\times}8$ partitions, each block can be further divided into partitions of $8{\times}4$, $4{\times}8$ or $4{\times}4$ luma samples. Independent translational MVs, and corresponding reference frame indexes, are assigned to each luma partition, meaning that each inter MB can be encoded using a number of MVs ranging from 1 (for a $16{\times}16$ partition) to 16, when $4{\times}4$ partitions are used. Each MV can be associated to a different reference frame, among a set of pictures stored in a dedicated buffer.

In *B* slices, inter prediction uses two lists of reference frames. Each motion predicted signal can be determined using a MV associated with list 0 or list 1, but it can also use a weighted average of one picture from list 0 and another from list 1. Furthermore, a *copy* or B_Skip mode is also supported, where both the prediction signal and used MV are inferred from the previous MBs' data, in a process similar to the one used by P_Skip MB's.

In most video coding standards, the MC prediction signal is used to determine a residue block that is encoded using a compression scheme, typically based on transform coding. This residual error coding step of the algorithm, also present in the implementation of the H.264/AVC reference software [3], is generally accepted as crucial for the performance of video encoders.

The impressive gains achieved by the motion compensation process of H.264/AVC, allied to the efficient RD optimisation scheme, led us to hypothesise that, at least in some cases, motion compensation is efficient enough to obviate the need of predicted residual coding, that introduces an additional bit-rate overhead. Experimental tests (see results in section 4) showed that in the case of B slices, overriding the coding of the MC residue allows for increased performance of an H.264/AVC based video encoder, at low to medium rates. The used process is described in the following section.

2.1. The Overriding Process

In order to create a compliant encoder, instead of simply removing all the residual error information from the bitstream, all samples of motion compensated residue block are set to zero, prior to the residual encoding step. The H.264/AVC encoding syntax efficiently compresses the overhead associated with these null coefficients through the use of the *Coded Block Pattern* (CBP) parameter and entropy coding. The existence of only null coefficients in the transformed MC residue block is signalled by the encoder by setting the CBP parameter to zero. This fact is efficiently exploited by the entropy coding tools defined in the H.264/AVC standard.

The original RD optimisation and mode assignment procedures are maintained, allowing for the choice of the best encoding mode for each block considering the null residue condition. The resulting bit-stream is fully compliant with the original H.264/AVC reference decoder. Also, because the direct and inverse transform steps are discarded for B slices, a reduction in the computational complexity of both the encoder and the decoder is achieved.

3. DICTIONARY-BASED VIDEO ENCODING

Dictionary-based source encoders divide the message into blocks that are replaced by codewords [4, 5, 6]. In the case of lossless coding, the codewords must be exact matches of the message block. Lossy encoders, on the other hand, chose the codeword in order to provide an accurate representation of the original message, but some level of distortion is accepted.

In spite of the great success achieved by these encoders in the compression of one-dimensional data (Unix Compress, ZIP, etc), there have been rare proposals to adapt these methods to two-dimensional data (images) and even fewer for video coding. Notable exceptions can be found in [7] and [8]. In this section we show that the proposed method of overriding motion-compensated residual coding in H.264/AVC can be regarded as a successful implementation of a lossy dictionary-based video encoder.

One important family of dictionary-based algorithms emerged from the work of Lempel and Ziv [4, 5]. In LZ77 [4] encoders, pointers are used to identify the longest match for the current block within a search buffer, composed by the most recently coded segments of data. In this case the length of the matched string is also transmitted. LZ78 [5] implementations use an explicit dictionary, composed by an indexed list of previously used portions of the message.

Another important class of dictionary-based algorithms is the one based on vector quantisation (VQ) [6], where a set of codewords is stored in a dictionary, that can either be fixed or adaptive. Each message block is thus encoded by replacing it with a dictionary index.

In the presented method, the best approximation of the current block, determined by the ME process, also corresponds to a previously encoded portion of the message (namely, a block, or combination of blocks, of the reference frames). We can thus think of the motion vectors as LZ pointers, that are encoded by H.264/AVC using a predictive scheme and a context adaptive arithmetic encoder. The length of the message used in each approximation is implicitly encoded by the partition size used in the MC process.

We can also relate the patterns stored in the reference frames to an adaptive dictionary. In this case, each motion vector acts as an index that identifies the chosen codeword. The use of different partition sizes in the MC process can be regarded as the use of several dictionaries, that store blocks with different dimensions. The dictionary indexing process is implicitly determined by the choice of the partition size.

The adaptation process of the dictionary consists of the use of a variable set of codewords, that are chosen according to temporal (related to the choice of reference frames) and neighbourhood (represented by the search window) criteria. This increases the dictionary's efficiency, because it tends to use codewords that are likely to be similar to the current frame's blocks. The use of B slices can be interpreted as an extension of the dictionary-based coding paradigm to the case were a weighted combination of two codewords is employed.

4. EXPERIMENT DESCRIPTION AND SIMULATION RESULTS

The implementation of the proposed method is based on the H.264/AVC reference software. Version *JM9.3* [3] was used for availability reasons, since the changes introduced in more recent versions are not relevant for the effects of our study. A set of experimental tests was performed using the first 100 frames of several well known test sequences *Foreman* (CIF),

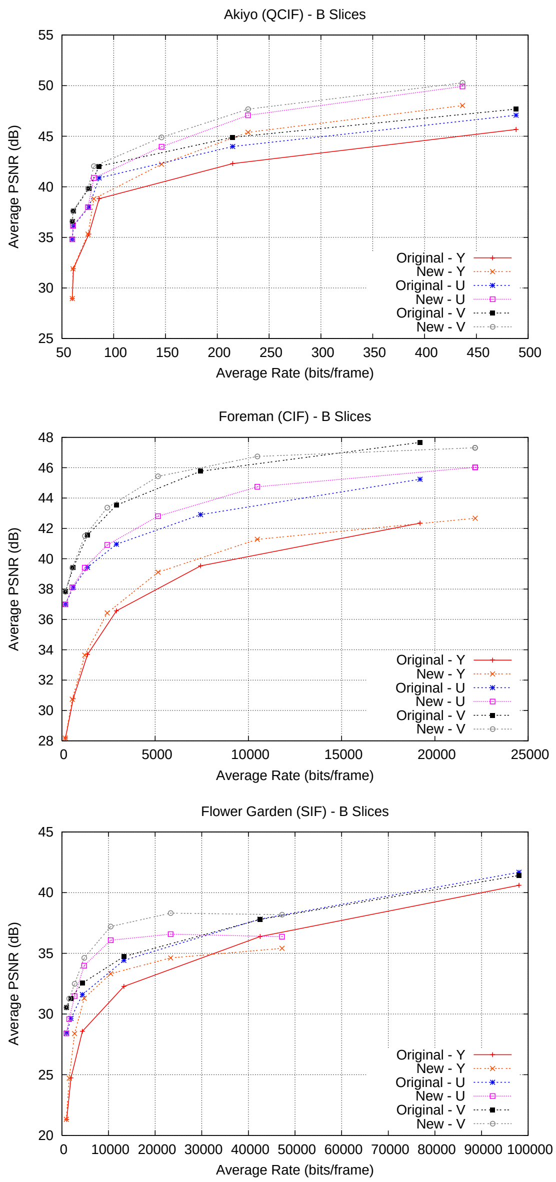

Fig. 1. Experimental results of the original H.264/AVC JM9.3 encoder and the new encoder.

Mobile and Calendar (CIF), *Akiyo* (QCIF) and *Flower Garden* (SIF), with 4:2:0 colour subsampling. These sequences were chosen due to their common use as benchmarks for video encoding tests and also because they represent a wide variety of video content: from a typical head and shoulders sequence to highly detailed sequences, with large motion.

Experimental tests were performed for a wide set of input parameter values, in order to assess the effect of each of these parameters in the performance of the proposed method. However, some configurations are common to all simulations, namely: RD optimisation is enabled, 30 Hz frame-rate, variable bit-rate (VBR) mode, only the first frame is intra, 5 reference frames are used, CABAC entropic coding and deblocking filter are enabled and single level bipredictive subpixel motion estimation was used.

Figure 1 shows the results of a comparative test were the *high* profile was used. The values of some other important parameters are: bi-predictive coding and intra MB in non-intra slices were activated, 8×8 block transform was enabled and an *IPBP* pattern was used. Residual error encoding was disabled only for non-intra blocks of *B* slices, with the coding for the I and P slices being left unaltered. Since we have observed that the results for the I and P frames are the same in the two cases, figure 1 only shows the RD curves for B slices. Different target bit rates were achieved by varying the values of the QP parameter according to the values of table 1. Note that there is a direct relationship between the values of QP and of the Lagrangean multiplier parameter λ, used in the RD optimisation.

In figure 1 we observe the gains attained by the proposed technique, for all components of the *B* slices of all tested sequences, specially for low to medium rates. In these cases the motion compensation procedure is able to efficiently reconstruct the bidirectionally predicted frames from the previously encoded slices, with no need to encode an additional residual error. For sequences with small motion, not using residual coding for *B* slices is always advantageous, achieving, in some cases, the same PSNR using less than half the bitrate of the original H.264/AVC encoder. For sequences with complex or high velocity motion, the method still outperforms the original encoder, but only for low to medium qualities/bit rates. A subjective analysis of the reconstructed sequences reveals that no noticeable artifacts are introduced by the process and that the subjective quality variations correspond to the ones of the PSNR value.

Figure 2 represents the total number of MB's encoded with each prediction mode for the *B* slices of sequences *Foreman* and *Mobile and Calendar* using $QP = 20$. We may observe that, even for this relatively low value of the QP parameter, the copy mode is used frequently. Also, when we deactivate the residual error encoding, the motion estimation uses more partitions of a small size (8×8 or smaller), that allow for smaller distortion values, favouring the RD performance of these modes when no residue is encoded. For the *Mobile and Calendar* sequence we can also observe an increase in the number of Intra predicted MB's. Intra blocks are used when ME alone is not able to efficiently reconstruct the original signal, penalising the value of the Lagrangean cost function for MC modes.

When no residual information is used, the encoder is able to compensate the losses by using smaller block sizes for motion compensation. Nevertheless, this process is limited by the minimum block size of 4×4 allowed by H.264/AVC. This means that for higher rates, the encoder is not able to provide

QP for B slices	45	40	35	30	25	20	15*
QP for I and P slices	43	38	33	28	23	18	13*

Table 1. Used QP values (*only with no residual coding).

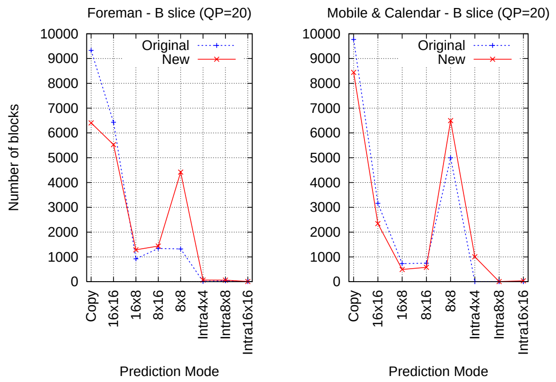

Fig. 2. Number of MB's that use each of the available prediction modes for the *B* slices.

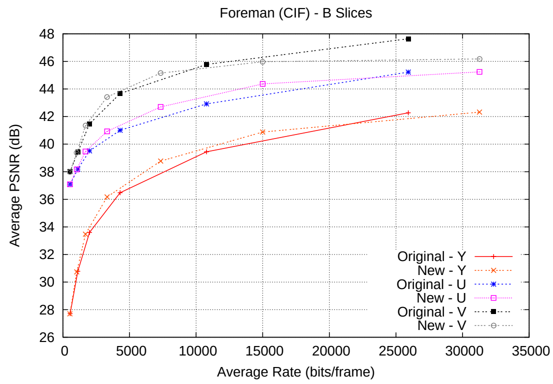

Fig. 3. Experimental results of original H.264/AVC JM9.3 and new encoder using *IPBBBP*... GOP, Foreman sequence.

the needed fidelity. In fact, the plots of figure 1 show that the PSNR of the B slices tends to be bounded for rates above a given threshold (for low values of QP); this phenomenon is more evident for sequences with higher motion. In these cases the use of residual error information is advantageous. From this one may conjecture that the use of smaller block sizes in the MC process may allow for good performances also in the high quality/bit rate regions.

In order to better evaluate the proposed technique, other tests were performed by varying the encoder's configuration. The relative gains presented previously were also observed when the *Main* profile of the H.264/AVC encoder was used, when the bi-predictive coding mode was turned off, and also when the width of the motion search window was increased to 32×32 pixels. For all tested cases, the proposed method maintained an advantage for all sequences/bit rate scenarios. Even when the number of B slices was increased (which compromises the accuracy of the prediction step), the same tendency has been observed. The difference is that the points for which the absence of residual data becomes a disadvantage happen sooner (lower rate value) than for the *IPBP* GOP. However, for low to medium rates it is still advantageous to override the transmission of residual data, as can be seen if figure 3 for the Foreman sequence using three B slices.

This process is not suitable to be used with P slices. This is so because of the inferior performance of unidirectional MC process (due to factors like problems with uncovered background, distance between key frames, etc.). Experimental results, not shown here, confirm this statement.

5. CONCLUSIONS

In this paper we investigated the performance of H.264/AVC when residual error encoding is deactivated for B slices. Experimental results demonstrated that this technique can provide significant improvements in the performance of the original encoder, specially at low to medium bit rates. Gains were observed for a wide range of test sequences and encoder configurations.

A formal analysis of this process relates it to coding paradigms not usually associated with video coding, like vector quantisation and Lempel-Ziv encoders. These results indicate that it may be worthy to further investigate the applicability of dictionary-based techniques to video coding.

6. REFERENCES

[1] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), *Draft of Version 4 of H.264/AVC (ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 part 10) Advanced Video Coding)*, March 2005.

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

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

[4] Jacob Ziv and Abraham Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[5] Jacob Ziv and Abraham Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, 1978.

[6] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Springer, 1991.

[7] M. de Carvalho, E. da Silva, and W. Finamore, "Multidimensional signal compression using multiscale recurrent patterns," *Elsevier Signal Processing*, , no. 82, pp. 1559–1580, November 2002.

[8] N. Rodrigues, E. da Silva, M. de Carvalho, S. de Faria, and V. Silva, "Improving H.264/AVC inter compression with multiscale recurrent patterns," *IEEE International Conference on Image Processing*, October 2006.