COMPARISON BETWEEN H.264/AVC AND JPEG2000 ARITHMETIC ENCODERS

Ottavio Campana, Edoardo Vaccaro

Dept. of Information Engineering, University of Padova
Via Gradenigo 6/B, 35131 Padova, ITALY

ABSTRACT

JPEG2000 Part 3 defines the MJ2 file format that should be used to encode video sequences as a succession of independently encoded images. Since MJ2 involves only intra-frame coding, performances of JPEG2000 cannot be easily compared with those of other codecs that exploit both intra and inter-frame coding, e.g the H.264/AVC standard. In this paper we present a hybrid encoder where the JPEG2000 EBCOT algorithm substitutes the standard quantization and arithmetic coding blocks of H.264/AVC. We used it compare the efficiency of the CABAC arithmetic encoder of H.264/AVC with the EBCOT algorithm of JPEG2000. The obtained results can give directions for next generation video coding standards.

Index Terms— Image coding, Multimedia computing, Video signal processing, Video codecs, Video reviews

1. INTRODUCTION

Video coding can take advantage of both spatial and temporal correlation to enhance compression performance. The former if known as *intra*-coding and correlation is exploited within each frame. The latter is known as *inter*-coding, and correlation is exploited between adjacent frames through motion compensation. An example of video codec using only intra-coding is JPEG2000 [1] Part3 (also known as Motion-JPEG2000), where every frame is compressed applying the JPEG2000 algorithm for still images. Instead, the state of the art for intra and inter video coding is the H.264/AVC [2] standard, which is based on a very fine motion compensation using variable size blocks.

The lack of inter-frame coding in JPEG2000 makes hard the comparison between algorithms and arithmetic encoders. Since prediction plays an important role in compression, performance of the Context-Based Adaptive Arithmetic Coding (CABAC) [3] and the one of the Embedded Block Coding with Optimized Truncation (EBCOT) [4] cannot be easily compared, because prediction in the two standards is very different. JPEG2000 and H.264/AVC were compared encoding video sequences with intra-prediction only, and discarding temporal prediction available in H.264/AVC. In [5], this comparison was proposed, and H.264/AVC performs better than JPEG2000, even using only a reduced subset of its features. However, this comparison methodology only shows the functioning of codecs as a whole, but no indications are given for single algorithms. Arithmetic encoding in H.264/AVC is based on Run-Length Encoding (RLE) and binarization, and on bitplanes encoding in JPEG2000. Thus, the comparison of these two approaches determines which strategy gives better compression ratios. In this paper we present a new coding scheme where a modified version of EBCOT, taken from JPEG2000 Kakadu reference code, is inserted into the JVT H.264/AVC reference encoder. Using this new encoder, it is possible to compare CABAC with EBCOT using both intra and inter prediction.

This article is organized as follows. In Section 2 the description of the new encoder is given. Experimental results of the comparison between the proposed encoder and the standard H.264/AVC one are presented in Section 3 and, finally, concluding remarks are given in Section 4.

2. SCHEME DESCRIPTION

We decided to implement this hybrid scheme as an incomplete extension of the bitstream syntax of the H.264/AVC standard. In fact, the standard gives the possibility of introducing a new coding routine for the residual information because it uses two different functions for CABAC and Context-Adaptive Variable Length Coding (CAVLC), respectively for arithmetic encoding and Huffman encoding. Adding a third entropy coding algorithm only requires a little change in the `Picture Parameter set` because the `entropy_coding_mode_flag` field has to be extended from one to two bits to select the desired entropy coding mode and the implementation of a new `residual()` function. These functionalities were implemented in our experimental encoder, only the Network Abstraction Layer (NAL) unit creation is missing. This is not required to evaluate how the new encoder works and for practical reasons compressed residual information was saved in a file separated from the standard one, holding intra and inter prediction.

Since JPEG2000 is based on the Discrete Wavelet Transform (DWT) and H.264/AVC is based on the 4x4 Discrete Cosine Transform (DCT), good performance of the arithmetic encoder in EBCOT cannot be guaranteed. In fact, the output of the two transforms is different and context models used for DWT cannot match the statistics of the DCT output. In order to make these two transforms interchangeable, coefficients taken from the 4x4 DCT blocks need to be reordered, obtaining a structure similar to the output of DWT and creating 16 bands. The reordering process, as shown in Figure 1, is obtained grouping each of the 16 coefficients of the 4x4 blocks near those of the other blocks sharing the same position inside them. Although DCT and DWT are very different, the reordered DCT output and the DWT output share some common characteristics: the lowest band of the reordered DCT and DWT, respectively the DC and LL band, have the same p.d.f. as the original image, while the remaining bands have a laplacian like p.d.f. Therefore, EBCOT context models can match the residual information transformed by the DCT. An example of reordered DCT coefficients is given in Figure 2.

In the first implementation of the coding scheme, we substituted the DCT, the quantization block and the entropy coding of H.264/AVC. The standard H.264/AVC DCT transform was replaced by a fixed point reversible DCT which operated on the residual information, after prediction. The transformed residual was subsequently compressed by EBCOT.

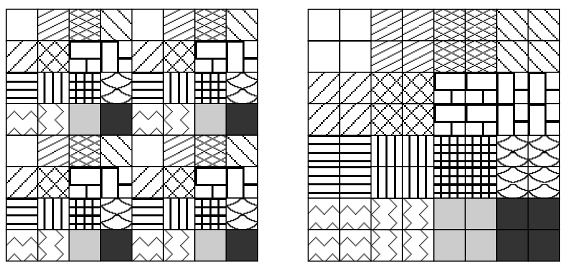

Fig. 1. Example of reordered transformed coefficients of four 4x4 blocks. Four 4x4 blocks output from the DCT are shown on the left, while on the right the reordered coefficients are shown. Each band is obtained grouping the coefficients of the four blocks that have the same coordinates in the original blocks.

The H.264/AVC modified DCT was replaced by a fixed point implementation because of the following reasons:

- Quantization was performed by EBCOT and therefore it needed to be separated from quantization in H.264/AVC. H.264/AVC transform does not scale the coefficients, instead this operation is embedded in the quantization block, in order to get a faster coding process. Thus, transformed coefficients are a scaled version of what they are expected to be and they cannot be directly given to EBCOT for compression.

- A reversible transform can be used.

- Using a fixed point implementation, the number of bit used in the representation can be chosen a priori and therefore it is possible to select the number of bitplanes that have to be encoded by EBCOT.

- Floating point implementation is not necessary because of arithmetic precision.

From a practical point of view, rewriting the DCT code was also necessary because JVT code uses functions and variables that cannot be easily used inserting EBCOT in the reference code.

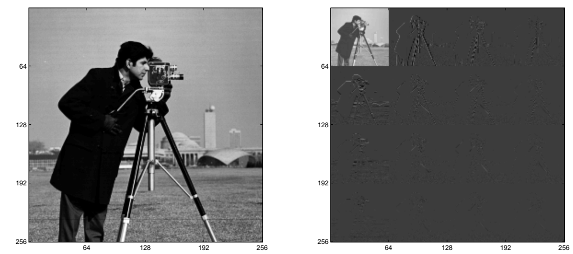

Fig. 2. Example of original image on the left and reordered DCT on the right. The 16 bands obtained reordering the DCT coefficients can be easily seen.

Under test, this solution gave poor results. In fact, this scheme treats all the macroblocks as if they had a non null prediction error, because all the residuals are directly passed to EBCOT, which wastes a lot of bits to encode prediction errors that have small energy. This problem is not present in H.264/AVC because, immediately after transform, quantization is performed. When the prediction error is nullified by quantization, the coding process needs only to set the `coded_block_pattern` to zero in order to skip the residual information coding process. Moreover, this scheme had an important drawback if compared with H.264/AVC: it broke Intra 4x4 prediction. In fact, when a macroblock in H.264/AVC is encoded using Intra 4x4 mode, all its 4x4 blocks are predicted from the adjacent already reconstructed ones. This is not possible using the proposed scheme, because the reconstructed image is available only when the whole frame is reconstructed and therefore spatial prediction cannot work as well as in H.264/AVC.

To solve this problem we decided to use H.264/AVC DCT and to let it quantize the error signal, obtaining a new scheme. With this approach, correctly predicted macroblocks can exploit the `coded_block_pattern` set to zero in order to be skipped, while only those with a significant prediction error have to be encoded. Since `coded_block_pattern` works on 8x8 submacroblocks, we decided to pack the non null submacroblocks into a rectangle with the same aspect ratio as the original image shifting them to the left of the rows and moving them to the right of the upper rows, if needed. When the number of submacroblocks is not enough to cover the rectangle, it is filled with null submacroblocks.

This new scheme has the advantage of creating the same frames that an H.264/AVC standard encoder would produce, thus the comparison between it and the reference H.264/AVC encoder developed by the Joint Video Team (JVT) can be easily done comparing the number of bits per pixel required. This approach, whose scheme is shown in Figure 3, gave very good results in term of rate-distortion, comparable with those of the standard H.264/AVC encoder.

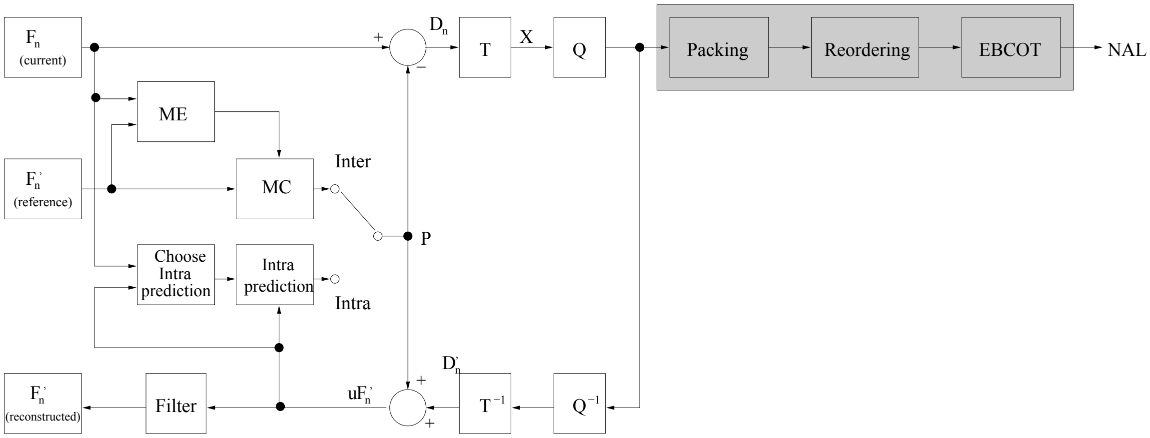

Fig. 3. Scheme of the modified hybrid H.264/AVC - EBCOT encoder. In the grey rectangular, the reordering and CABAC entropy coding blocks of H.264/AVC were substituted by 8x8 the submacroblocks packing and reordering blocks and EBCOT.

3. EXPERIMENTAL RESULTS

We ran several simulations to compare the standard H.264/AVC encoder with the proposed schemes. The H.264/AVC configuration is summarized in Table 1, and the proposed schemes used the same settings, where applicable. We tested the schemes with the "*Foreman*", "*News*" and "*Mobile*" QCIF and CIF sequences at 25 frames per second using a Group of Pictures (GOP) of 16 frames, inserting two B frames between I and P frames.

As previously said, under test the first scheme had a disappointing behavior: the obtained Rate-Distortion (RD) curve was not monotone, as it was expected to be. This is due to the fact that for low quality sequences EBCOT is not able to efficiently compress the residual information, which consists at most of zeros. Not only this appears in the poor performance of the first scheme for small and medium values of QP, but for big values of QP the arithmetic encoder has to work with data that does not match its contexts wasting lot of bits, and the RD curve looses its monotonicity. The RD curve of the first scheme and of the standard H.264/AVC is shown in Figure 4. This inefficient behavior is particularly evident with static video sequences as the "*News*" CIF sequence. This happens because the EBCOT must inefficiently encode a lot of null symbols.

The second encoder was able to perform as well as the standard H.264/AVC encoder for big and medium values of QP. Outperforming compression ratios were registered for high quality video sequences. In fact, for $QP = 0$ it uses the half of the bits per pixel of the H.264/AVC encoder. In Figures 5 and 6, the RD curves respectively for the "*Mobile*" and "*Foreman*" CIF sequences are shown for both the standard H.264/AVC and the second scheme. Skipping the null 8x8 submacroblocks, EBCOT is able to achieve higher performance than CABAC.

These outstanding results suggest a couple of reflections. The arithmetic encoder of EBCOT is not able to efficiently compress regions where most of coefficients are null. EBCOT relies on a MQ encoder which works on bitplanes, while H.264/AVC encodes the coefficients after their binarization using RLE to skip null symbols. This explains why the first scheme is not able to achieve performance comparable with the ones of CABAC: null coefficients require a lot of bits even though they do not carry information. When the number of null coefficients is small, as in the second scheme, EBCOT is close to CABAC if not even better.

Parameter	Value
Profile	Main
Intra prediction modes	all
Inter prediction modes	all
Entropy Coding	CABAC
8x8 transform	no
RD-opt	no
Rate Control	no
Deblocking Filter	yes

Table 1. Parameters shared between the standard H.264/AVC encoder and the proposed schemes.

A second possible consideration is about CABAC. The adoption of arithmetic encoding in video coding standards is recent: it was firstly inserted into H.263 annex E, and subsequently in H.264/AVC to achieve better compression rates than Huffman encoding (i.e. CAVLC in H.264/AVC or similars in previous standards). Since EBCOT is able to achieve better performance than CABAC by halving the bits required for high quality sequences, many improvements can be done in contexts modeling, coefficients binarization and coefficients scan. Future video coding standards have to focus on their arithmetic encoder to fully exploit the potentials offered by this newly adopted technique. This assumption was confirmed also in [6], where a better choice of contexts gave a 10% bit-

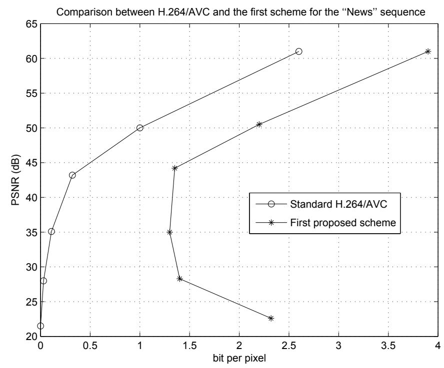

Fig. 4. Comparison between the first scheme and the standard H.264/AVC for the "*News*" sequence. Highlighted points for H.264/AVC correspond to $QP \in \{0, 10, 20, 30, 40, 50\}$, while points for EBCOT were obtained truncating the bitstream and imposing the same length of the homologous H.264/AVC NAL units length.

stream reduction in the CABAC performance.

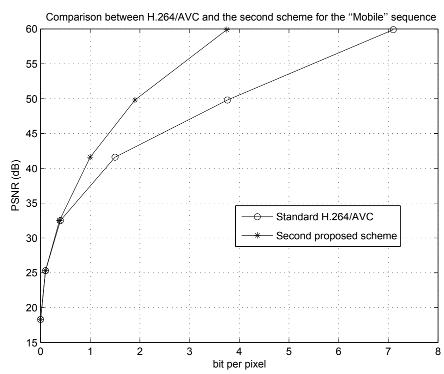

Fig. 5. Comparison between the second scheme and the standard H.264/AVC for the "*Mobile*" sequence. Highlighted points correspond to $QP \in \{0, 10, 20, 30, 40, 50\}$.

4. CONCLUSIONS

In this article a new hybrid video encoder was presented, based on the standard H.264/AVC encoder with the insertion of the JPEG200 EBCOT algorithm. We used this encoder to compare arithmetic coding performed by CABAC in H.264/AVC and by EBCOT in JPEG2000.

Performance of this hybrid encoder can significantly vary, depending on how quantization is performed. When quantization is made by H.264/AVC and not null 8x8 submacroblocks are packed together into a rectangle with the same aspect ratio of the original sequence, EBCOT is able achieve significantly higher performances than CABAC. When quantization is performed by EBCOT more bits per pixel are required and the RD function looses its monotonicity because input values do not fit EBCOT context models.

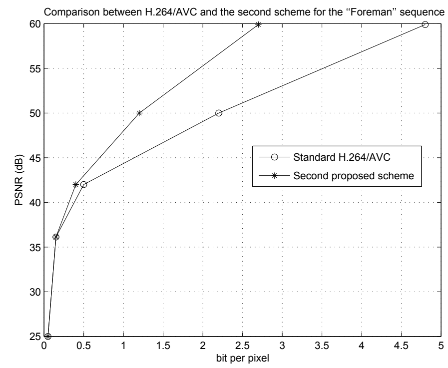

Fig. 6. Comparison between the second scheme and the standard H.264/AVC for the "*Foreman*" sequence. Highlighted points correspond to $QP \in \{0, 10, 20, 30, 40, 50\}$.

Since processors in mobile devices and set-top-box have now enough computational power, arithmetic encoders are becoming popular and therefore particular attention has to be focused on their design. Particularly, two are the most important aspects of encoders that need further investigation: context building and coefficients scanning, by bitplanes or by run-length encoding.

5. REFERENCES

[1] D. S. Taubman and M. Marcellin, *Jpeg 2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2002.

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

[3] H. Schwarz D. Marpe and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, 2003.

[4] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

[5] H. L. Cycon D. Marpe, V. George and K.U. Barthel, "Performance evaluation of Motion-JPEG2000 in comparison with H.264/AVC operated in pure intra coding mode," *SPIE Conf. on Wavelet Applications in Industrial Processing, Photonics East, Rhode Island, Oct*, 2003.

[6] S. Milani and G.A. Mian, "An improved Context Adaptive Binary Arithmetic Coder for the H.264/AVC standard," in *Proc. of the 14th European Signal Processing Conference*, Firenze, Italy, Sept. 2006.