H.264/AVC FINE GRAIN SCALABILITY USING BITPLANE CODING

João Ascenso
joao.ascenso@lx.it.pt
Escola Superior de Tecnologia de Setúbal/Instituto de Telecomunicações
Campus IPS, Estefanilha, 2910-761 Setúbal, Portugal.

Fernando Pereira
fp@lx.it.pt
Instituto Superior Técnico/Instituto de Telecomunicações
Av. Rovisco Pais, 1049-001 Lisboa, Portugal.

ABSTRACT

The H.264/AVC standard represents an important step in the evolution of video coding standards since it offers a significant improvement in terms of rate-distortion efficiency by providing more than a factor of two in bit-rate savings against the popular MPEG-2 Video standard. In this work, a new video coding scheme which combines the fine grain scalability properties of the MPEG-4 FGS standard and the recent advances in the area of nonscalable video coding represented by the H.264/AVC standard is presented and evaluated. This new video coding scheme, called AVC-FGS, uses a H.264/AVC conformant base layer; in the enhancement layer the tools of the base layer are reused as much as possible in order to share a maximum number of tools between layers and to benefit from the H.264/AVC encoding tools properties, e.g. high coding efficiency, low complexity integer transforms.

1. INTRODUCTION

As several experiments have shown, the current MPEG-4 FGS (Fine Grain Scalability) coding scheme [1] shows a low coding efficiency regarding the corresponding non-scalable single layer encoding for a wide range of bitrates and qualities [2]. Among other reasons, the MPEG-4 FGS base layer performance limits the overall FGS coding efficiency. So, the use of the more efficient H.264/AVC (Advanced Video Coding) [3] encoder in the base layer is expected to bring improvements when compared with MPEG-4 ASP (Advanced Simple Profile) which is the MPEG-4 FGS base layer coding solution.

In the literature there has been some work reported regarding the combination of the MPEG-4 FGS and the H.264/AVC standards [4] [5]. The approach taken is a direct implementation of the MPEG-4 FGS enhancement layer, without any modification, on the top of the H.264/AVC encoder. However, this scalable structure has several drawbacks, mainly because it introduces new encoding and decoding tools in the enhancement layer, for functionalities already present in the base layer. For example, with this straightforward implementation, it is necessary to implement two different transforms: in the enhancement layer, the 8×8 pixels Discrete Cosine Transform (DCT) and in the base layer the H.264/AVC 4×4 pixels Integer DCT transform [3]. The entropy coding scheme used in MPEG-4 FGS (Huffman coding) needs also to be implemented, since H.264/AVC uses new entropy coding schemes. The burden of having duplicate tools for the same functionality would limit the adoption of this type of H.264/AVC scalability because it significantly increases the complexity of both the encoder and decoder. In addition, this scalable structure would not take advantage in the enhancement layer of the new tools present in the H.264/AVC standard, e.g. increased efficiency, no transform mismatch error.

In this paper, a new scalable video coding scheme, called AVC-FGS, is proposed; this solution is based on the MPEG-4 FGS bitplane coding tool [1], and some of the tools specified in the recent H.264/AVC standard [3]. The AVC-FGS architecture proposed here maintains all the FGS capabilities intact, notably fine grain adaptation to dynamic changes in network conditions, low complexity decoding and memory requirements, and packet loss resilience. In the AVC-FGS enhancement layer, the tools in the H.264/AVC base layer are reused as much as possible to take maximum advantage of them. The first step is a careful selection of the base layer tools to use in the enhancement layer: the transform and the entropic coding tools are the most important in the context of the FGS-like bitplane coding adopted here to provide the fine grain scalability.

This paper is organized as follows. First, Section 2 presents some of the H.264/AVC tools to be used in AVC-FGS; then Section 3 presents in detail the proposed AVC-FGS codec architecture. In Section 4, the AVC-FGS efficiency is evaluated in comparison with the MPEG-4 FGS solution. Finally, conclusions and some future work topics are presented in Section 5.

2. BRIEF DESCRIPTION OF H.264/AVC TOOLS

The H.264/AVC standard uses a 4×4 integer transform, and a 2×2 or 4×4 Hadamard transform. Both transforms can be exactly computed by means of integer arithmetic, thus avoiding inverse transform mismatch problems. They can also be computed without multiplications, just additions and shifts if 16-bit arithmetic is used, thus reducing the computational complexity. The smaller block size transforms, comparing to previous coding standards, are compatible with the H.264/AVC finest motion compensation (4×4 blocks) and also yield a reduction in ringing artifacts.

The entropy coding techniques defined in the H.264/AVC standard are CA-VLC (Context Adaptive Variable Length Coding), CABAC (Context Adaptive Binary Arithmetic Coding) and UVLC (Universal Variable Length Encoding). Both CA-VLC and CABAC can achieve a better efficiency performance than UVLC but come at the expense of higher complexity. The UVLC entropy coding works with variable length exp-Golomb codes with symmetric and regular structure. This allows a simplification of the encoding and decoding processes and the use of a single table to map all the syntax elements into UVLC codewords.

3. AVC-FGS ARCHITECTURE

The proposed AVC-FGS encoder architecture is shown in Figure 1. The overall coding architecture is similar to the one used in MPEG-4 FGS, with two separate layers. The base layer uses a H.264/AVC encoder conformant to the Baseline profile;

this solution provides a good quality with a relatively low complexity. The enhancement layer provides fine granularity scalability through bitplane encoding; this means the enhancement layer bitstream may be truncated at any point after the encoding is complete.

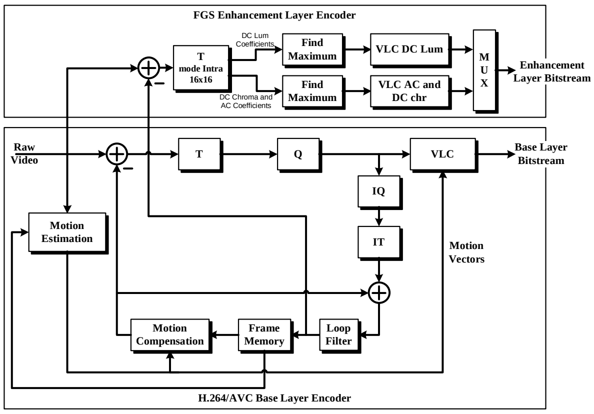

Figure 1 – AVC-FGS encoder architecture.

The corresponding decoder is able to reconstruct the video from the base layer and the truncated enhancement layer bitstream; the final quality obtained depends on the number of bits received and decoded, notably from the enhancement layer. The AVC-FGS enhancement layer encodes a residual image which corresponds to the difference between the original image and the base layer decoded image.

Like in the MPEG-4 FGS standard, the enhancement layer information is not used in the base layer motion prediction loop; this means the base and the enhancement layers are only predicted from references in the base layer. Since the prediction is always based in the lower quality base layer reference, the AVC-FGS coding efficiency suffers a loss when compared to the nonscalable H.264/AVC scheme. However, this type of architecture offers excellent error recovery capabilities when data losses or errors occur in the enhancement layer since no error propagation occurs. Also if the decoder only receives part of the enhancement layer, no drift effect occurs at all and thus it is possible to perform a fine adaptation to the bandwidth changes. In order to benefit from the more efficient H.264/AVC coding tools and avoid duplication of tools for similar functionalities in the base and enhancement layers, the architecture proposed here enables fine grain scalability in the context of the H.264/AVC standard by implementing some of its key tools to encode the enhancement layer as well, namely:

1) Transform: The enhancement layer encoder uses the same base layer transform and classifies all macroblocks as Intra (mode Intra16×16) which means that no prediction is performed to code the residue (all enhancement layer images are Intra). With this mode, all 16 4×4 luma blocks in a macroblock are first transformed with the Integer DCT transform. Then the DC coefficients of the 4×4 blocks are transformed again using a 4×4 Hadamard transform. For chroma blocks, the process is similar; the only difference is the size of the Hadamard transform (2×2), since the number of blocks for each color component (U or V) is now 4 (4:2:0 subsampling). This 2-level transform is usually referred as a hierarchical transform and was chosen mainly because it explores the correlation among the DC coefficients of neighboring blocks. This spatial correlation is typical in blocks with mostly flat pixel values, like some parts of the enhancement layer residual images. This was verified by means of a statistical study of the DC coefficients distribution.

2) Separate encoding of DC luma coefficients: After the transform is applied, there are three types of coding blocks: a) *DCLum*– DC luma 4×4 blocks; b) *DCChr* – DC chroma 2×2 blocks; and c) AC coefficients 4×4 blocks. In the enhancement layer, all *DCLum* blocks which belong to a bitplane are grouped and transmitted together to the decoder, i.e. one or more bitplanes of the *DCLum* blocks are transmitted before the remaining coefficients. The bitstream syntax takes into account the (perceptual) importance of each coefficient: within a bitplane, all DClum coefficients are sent first, and then, for each macroblock, the remaining coefficients are sent according to the following order: ACLum (16 blocks), DCChr (2 blocks) and ACChr (8 blocks). The enhancement layer bitstream syntax is structured in two levels: **FGSDCLumBitplane,** and **FGSDCChrACBitplane** in order to support this bitstream structure, significantly different from the MPEG-4 FGS standard.

3) Only UVLC entropy coding: The UVLC entropy coding scheme was the chosen entropy coding scheme to be used in the enhancement layer from the two entropy encoders used in the H.264/AVC Baseline profile base layer (CA-VLC and UVLC). In this profile, UVLC encodes all information except the transform coefficients which are encoded with CA-VLC. UVLC was the single entropy coding solution adopted for the AVC-FGS enhancement layer due to its simplicity and reduced complexity. UVLC codes are variable length codes with a regular construction: each code is made of a suffix and a prefix that includes a separating bit with value ‘1’. The bits of the prefix always have value ‘0’ and the bits of the suffix are used in the calculation of the corresponding codeword. For any codeword, the number of bits in the suffix is equal to the number of bits in prefix minus 1. The DC coefficients are encoded independently of the AC coefficients in order to match the statistical distributions of each type of coefficients. Finally, the DC coefficients are multiplexed with the remaining coefficients, in the same bitstream.

The architecture of the decoder follows from the architecture of the encoder and is presented in Figure 2. The base and enhancement layers are independently decoded; the base layer decoded image is added to the enhancement layer decoded residue (which depends on the amount of bits received) to obtain the final decoded video with better quality, of course always depending on the available bandwidth.

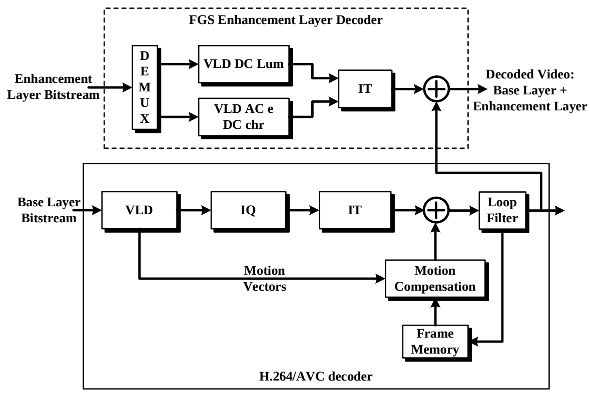

Figure 2 – AVC-FGS decoder architecture.

In the AVC-FGS decoder, the inverse operations of the encoder are performed. All the coefficients are entropy decoded (VLD), and the result is converted to the spatial/temporal domain with the inverse transform. Since the inverse transform defined in the H.264/AVC standard combines the normalization and quantification steps, the enhancement layer uses the scaling/quantification factors for the minimum quantization step (0); this corresponds to the case where less quantization is performed since the quality will ultimately depend on the number of bits received and thus no quantization has to be applied targeting a specific bitrate.

4. EXPERIMENTAL RESULTS

To evaluate the coding efficiency of the proposed AVC-FGS scheme, extensive experiments have been performed. In these experiments three codecs were used: a) **Codec 1:** MPEG-4 FGS encoder and decoder, which corresponds to the MPEG-4 Reference Software (part 5) [7]; b) **Codec 2:** H.264/AVC base layer encoder and decoder with a solution in the enhancement layer fully compliant with the MPEG-4 FGS standard (this means no H.264/AVC tools at all in the enhancement layer) and c) **Codec 3:** the AVC-FGS encoder proposed in Section 3. The last two codecs were developed by the author, and both are based on the H.264/AVC Joint Model 5.0 with rate control in the base layer [6]; only the Baseline profile tools were used in all experiments. Two sets of experiments were performed:

1) Base Layer Test: The main goal of this test is to evaluate the coding efficiency gain when the base layer uses the more efficient H.264/AVC standard (in comparison with the Advanced Simple Profile used in the MPEG-4 FGS standard). In this test, the enhancement layer of both codecs under test is the same, conformant with the MPEG-4 FGS standard. The test compares Codec 1 that uses in the base layer the MPEG-4 Advanced Simple Profile and Codec 2 that uses in the base layer the H.264/AVC standard.

2) Enhancement Layer Test: The main goal of this test is to find the relative performance of the AVC-FGS enhancement layer in comparison with MPEG-4 FGS always using the H.264/AVC standard (Baseline profile) in the base layer. While both codecs under test use a H.264/AVC base layer, the enhancement layer corresponds to the MPEG-4 FGS standard (Codec 2) and the proposed AVC-FGS solution (Codec 3).

The test conditions used are similar to the conditions defined by the MPEG group to evaluate the FGS technology [8]. In these test conditions, six scenarios (S1 to S6) are defined including a wide range of bitrates, and spatial and temporal resolutions. A large number of sequences with different motion and texture characteristics were chosen to represent various types of content, from the almost still "Boat" sequence to the extremely rapid "Rugby" sequence.

For each pair scenario/sequence, the corresponding scalable bitstream was truncated and decoded at several points in the specified bitrate range [R_b, R_{max}]; in these conditions, a PSNR versus bitrate curve was obtained. To evaluate the performance of the encoders for each test configuration, the Bjontegaard measures [9] were used to express the average difference between two PSNR curves. With this method, two values are obtained: a) dPSNR: average PSNR difference in dB over the range of bitrates defined for each scenario; and b) dRate: average bitrate difference in % over the whole range of PSNR. In Table 1, the results for each test configuration and sequence are presented.

[Rb, Rmax] in kbit/s	**Base Layer Test: Enhancement Layer: MPEG-4 FGS; Base Layer: H.264/AVC vs. MPEG-4 ASP**									
	BOAT		CANOA		RUGBY		STEFAN		TABLE TENNIS	
	dPSNR	dRate	dPSNR	dRate	dPSNR	dRate	dPSNR	dRate	dPSNR	dRate
S1: [16, 64]	4.767	99.98	0.681	15.41	0.301	6.75	0.016	1.69	2.445	47.43
S2: [32, 128]	4.017	99.95	0.931	24.75	0.933	21.95	0.691	25.16	2.660	62.54
S3: [64, 256]	6.465	99.83	3.086	54.63	3.027	52.24	3.011	61.07	3.914	77.75
S4: [128, 512]	2.025	71.56	1.829	42.70	1.795	37.73	2.446	56.92	3.083	69.37
S5: [256, 1024]	3.850	99.99	1.113	42.24	1.605	44.54	2.494	71.40	2.341	77.81
S6: [512, 2048]	2.258	75.05	2.212	45.64	2.523	47.60	2.311	51.64	2.047	61.21
Average	**3.897**	**91.06**	**2.649**	**50.135**	**2.775**	**49.92**	**2.566**	**60.26**	**2.748**	**66.02**
	Enhancement Layer Test: Base layer: H.264/AVC; Enhancement Layer: MPEG-4 FGS vs. AVC-FGS									
S1: [16, 64]	0.099	6.26	0.259	5.58	0.313	6.70	0.327	9.96	0.322	8.49
S2: [32, 128]	0.051	4.89	0.353	9.96	0.247	6.36	0.275	10.29	0.152	6.25
S3: [64, 256]	0.119	7.04	0.324	8.71	0.395	8.40	0.392	10.20	0.162	5.53
S4: [128, 512]	0.123	8.96	0.320	9.61	0.377	9.39	0.319	10.77	0.180	8.52
S5: [256, 1024]	0.038	4.22	0.132	5.03	0.202	6.66	0.148	7.45	0.077	5.68
S6: [512, 2048]	0.075	5.76	0.348	10.51	0.399	11.00	0.276	9.50	0.148	8.47
Average	**0.084**	**6.19**	**0.289**	**8.23**	**0.322**	**8.08**	**0.289**	**9.69**	**0.174**	**7.16**

Table 1 - Luminance test results for a wide range of scenarios and sequences.

1) Base Layer Test: Table 1 shows that for all combinations, the coding efficiency is higher when using the H.264/AVC standard in the base layer. As expected, the base layer has a significant impact on the global performance of FGS systems, with more significant gains for sequences with high correlation between frames (e.g. "Boat" and "Table Tennis"). The larger gains occur when the base layer has higher performance, mainly due to the H.264/AVC motion compensation tools that explore the temporal correlation in a very efficient way. In Table 1, some cases are marked with underlined blue letter since they correspond to cases where the MPEG-4 ASP encoder didn't manage to achieve the desired bitrate; this was due to the rate control used (TM5) and the spatial and temporal resolutions defined for each scenario. These results were not used in the calculation of the averages.

In conclusion, the use of the more efficient H.264/AVC encoder in the base layer allows a higher encoding efficiency for all scenarios with PSNR gains between about 2.6 and 3.9 dB.

2) Enhancement Layer Test: For the enhancement layer test, the AVC-FGS solution proposed in this paper has a coding efficiency slightly inferior to MPEG-4 FGS with H.264/AVC in the base layer. The main reason for this efficiency loss is the performance of the exp-Golomb codes in the UVLC technique. Since a single code is used to capture the statistics of all syntax elements for all bitplanes, it is reasonable to expect some loss when using the UVLC technique when compared to the Huffman coding in the MPEG-4 FGS, which uses different tables for each bitplane level. This fact was also observed by the VCEG group [10] and is more significant for small quantizers (QP<13). The exp-Golomb codes can be improved since the codewords are not designed according to the symbol probabilities, but simply assigned according to the fixed construction rule. For AVC-FGS, this effect is more evident when the number of symbols to encode is high (typically for higher bitrates), this means when the least significant bitplanes are decoded; Figure 3 illustrates this effect where up to 0.14 dB in coding loss occurs (at about 1 Mbit/s).

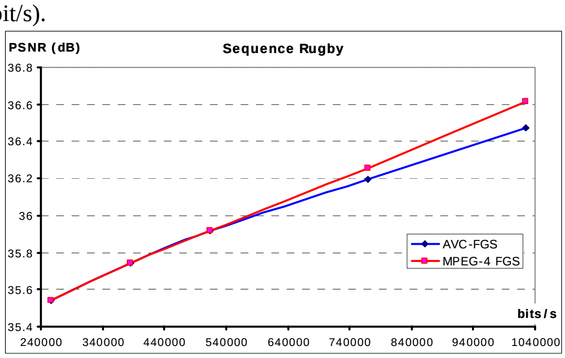

Figure 3 – Performance of AVC-FGS and MPEG-4 FGS with H.264/AVC in the base layer.

Also, this loss of efficiency is higher for sequences with low correlation between frames: e.g. for the sequence "Rugby", the loss is 0.2 to 0.4 dB. For this type of sequences (high motion activity), the quality of the base layer is lower, and thus the energy of the residue is high, which means that there are more bitplanes to encode in the enhancement layer. This leads to a loss of efficiency when the UVLC technique is used in the enhancement layer, since there are a larger number of symbols to encode and in this case the exp-Golomb codes do not match the symbols probability distribution in an efficient way.

5. FINAL REMARKS AND FUTURE WORK

In this paper a new solution for fine grain scalability based on the H.264/AVC standard is proposed, called AVC-FGS. The AVC-FGS coding scheme retains the MPEG-4 FGS functionalities, such as adaptation to dynamic changes in network conditions, packet loss resilience, etc. but takes also benefit of the higher coding efficiency provided by the new tools in the H.264/AVC standard.

The experimental results show that the use of the H.264/AVC in the base layer can improve the coding efficiency up to 3.9 dB in average PSNR over the MPEG-4 FGS scheme (MPEG-4 ASP in the base layer). It was also verified that the AVC-FGS enhancement layer shows a slight decrease in coding efficiency in relation to the MPEG-4 FGS like solution with an H.264/AVC base layer. However, is necessary to emphasize the lower complexity of the AVC-FGS enhancement layer in comparison with MPEG-4 FGS, especially due to the transform and entropy coding tools selected. This lower complexity will facilitate the future introduction of motion compensation tools in the enhancement layer (to increase the enhancement layer coding efficiency even if at the cost of some error resilience and drift problems) without sacrificing too much the global complexity of the system. Another topic for future research is the integration of the CA-VLC and CABAC tools in the AVC-FGS base layer which will cause an improvement of the coding efficiency, especially for the higher bitrates and some types of sequences.

ACKNOWLEDGMENT

The authors wish to acknowledge the support provided by the European Network of Excellence VISNET.

6. REFERENCES

[1] ISO/IEC International Standard 14496-2:2001/Amd 2, "Information Technology – Coding of Audiovisual Objects – Part 2: Visual, Amendment 2: Streaming Video Profile".

[2] H. M. Radha, M. van der Schaar, Y. Chen, "The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Over IP", IEEE Transactions on Multimedia, Vol. 3, No. 1, March 2001.

[3] ISO/IEC International Standard 14496-10:2003 "Information Technology – Coding of Audiovisual Objects – Part 10: Advanced Video Coding".

[4] H.-C. Huang, C.-N. Wang, T. Chiang, H.-M. Hang, "H.26L-based Robust Fine Granularity Scalability (RFGS)", ISO/IEC JTC1/SC29/WG11 M8604, Klagenfurt MPEG Meeting, July 2002.

[5] Y. He, F. Wu, S. Li, Y. Zhong, S. Yang, "H.26L-based Fine Granularity Scalable Video Coding", IEEE International Symposium on Circuits and Systems, Vol. 4, pp. 48-551, Phoenix - USA, May 2002.

[6] K. Suehring, "H.264/AVC Reference Software", JM 5.0, http://bs.hhi.de/~suehring/tml/.

[7] ISO/IEC International Standard 14496-5: 2001/Amd 1:2002, "Information Technology – Coding of Audiovisual Objects – Part 5: Reference Software".

[8] Video Group, "FGS Core Experiments", ISO/IEC JTC1/SC29/WG11 N2814, Vancouver MPEG Meeting, July 1999.

[9] G. Bjontegaard, "Calculation of Average PSNR Differences Between RD-curves", Doc. VCEG-M33, Austin - USA, April 2001.

[10] B. Jeon, "Entropy Coding Efficiency of H.26L", Doc. Q15-J57, Osaka VCEG Meeting, May 2000.