H.264/MPEG4-AVC Fidelity Range Extensions: Tools, Profiles, Performance, and Application Areas

Detlev Marpe and Thomas Wiegand
Fraunhofer Institute for Telecommunications –
Heinrich Hertz Institute (HHI)
Image Processing Department, Einsteinufer 37
D-10587 Berlin, Germany
[marpe,wiegand]@hhi.fraunhofer.de

Stephen Gordon
Broadcom Cooperation
Consumer Electronics Group
200 Brickstone Sq.
Andover, MA 01810, USA
sgordon@broadcom.com

***Abstract*—The technical work on the first amendment of the H.264/MPEG4-AVC video coding standard has recently been completed. In these so-called Fidelity Range Extensions (FRExt) a set of new coding tools is specified which is primarily targeted at providing significant improvements in coding efficiency for higher-fidelity video material. This paper presents an overview of the corresponding methods, briefly discusses some important aspects regarding profiles and applications, and finally provides experimental results for a performance comparison with existing coding technology.**

***Keywords*–H.264, MPEG4, Advanced Video Coding (AVC), Fidelity Range Extensions (FRExt), video coding, standards, JVT.**

I. INTRODUCTION

The technically aligned specification of ITU-T Recommendation H.264 and ISO/IEC MPEG4-AVC (Advanced Video Coding) [1], abbreviated as *H.264/MPEG4-AVC*, is the latest international video coding standard. As the result of the collective efforts of experts in the *Joint Video Team* (JVT) of ITU-T VCEG and ISO/IEC MPEG, H.264/MPEG4-AVC represents the most significant improvement in coding efficiency since the formerly and jointly developed video coding standard H.262/MPEG-2 Video [2]. Recently published results [3] indicate that H.264/MPEG4-AVC compliant encoders can achieve the same reproduction quality at half or less of the bit rate required by state-of-the-art MPEG-2 compliant encoders.

Technically, the design of the H.264/MPEG4-AVC video coding layer is based on the traditional hybrid concept of block-based motion-compensated prediction (MCP) and transform coding. Within this framework, a number of important innovative ideas have been developed. Some of these key features are given as follows (for more details, please refer to [4]):

- Enhanced MCP capabilities
- Multiple reference pictures and generalized B pictures
- Spatial intra prediction
- Small block-size transform in integer precision
- Content-adaptive in-loop deblocking filter
- Enhanced entropy coding methods

The first version of H.264/MPEG4-AVC, finalized and approved in 2003, was intended to be applied to a wide spectrum of applications ranging from mobile phones to digital cinema. By taking into account different requirements and limitations of typical target applications, three subsets of coding tools were defined as *Baseline*, *Extended*, and *Main* profile in the initial, i.e., 2003 approved H.264/MPEG4-AVC standard. In addition, a set of 15 *Levels* is specified in Annex A of the first version of H.264/MPEG4-AVC in order to provide conformance points with certain constraints on parameters such as, e.g., picture size, bit rate or frame rate [1].

After completion of the technical work on version 1 of H.264/MPEG4-AVC in March 2003, the JVT issued a Call for Proposals for support of extended sample bit depth and chroma format in the H.264/MPEG4-AVC standard. This standardization initiative was motivated by the rapidly growing demand for coding of higher-fidelity video material, especially in application areas like professional film production, video post production, or high-definition TV/DVD. The technical work on these so-called *Fidelity Range Extensions* (FRExt) of H.264/MPEG4-AVC was completed in July 2004, and the corresponding final draft amendment text was released in September 2004 [5]. Apart from the new functionalities provided by the FRExt amendment, further improvements in coding efficiency relative to the first version of H.264/MPEG4-AVC were demonstrated by using parts of the FRExt tool set within application domains that were initially targeted by the H.264/MPEG4-AVC Main Profile.

In this paper, an overview of the main distinguishing features of the H.264/MPEG4-AVC FRExt amendment is given. The next section describes the new FRExt coding tools with a particular emphasis on the 8×8 intra prediction and 8×8 transform. Section III presents the new profile definitions for FRExt and highlights some prospective application areas. In Section IV, some representative simulation results are provided for a performance comparison with existing standardized image and video-coding technology.

II. FREXT CODING TOOLS

In this section, we describe those coding tools in more detail that make the FRExt amendment a preferable choice for high-quality broadcast and entertainment applications. In addition, we briefly list all further FRExt specific coding tools, most of them primarily targeted at providing extended functionality in terms of sample bit depth and color format. The main difference between FRExt and non-FRExt H.264/MPEG4-AVC coding is the use of an 8x8 transform in addition to the 4×4 transforms.

0-7803-9134-9/05/$20.00 ©2005 IEEE

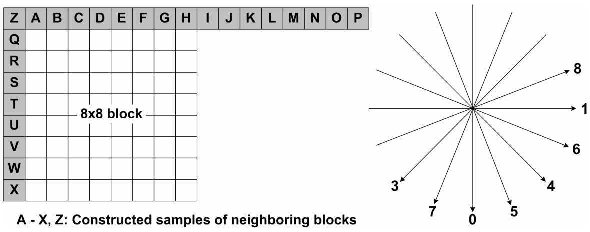
Z A B C D E F G H I J K L M N O P
Q
R
S
T
U
V
W
X
8x8 block
A - X, Z: Constructed samples of neighboring blocks
8
1
6
4
5
0
7
3

Figure 1: Left: Samples used for 8×8 spatial luma prediction. Right: Directions of spatial luma prediction modes 0, 1 and 3–8.

A. *8×8 Intra Spatial Prediction*

Intra prediction in H.264/MPEG4-AVC is conducted by using spatially neighboring samples of a given block, which are already transmitted and decoded. In version 1, two different types of intra prediction for luma values are specified: one type performing a prediction for a whole 16×16 macroblock (MB), and a second type for prediction of 4×4 blocks [1][4].

In the FRExt amendment, an intermediate prediction block size of 8×8 was introduced for spatial luma prediction by extending the concepts for 4×4 intra prediction in version 1. As illustrated in Figure 1, the luma values of each sample in a given 8×8 block are predicted from neighboring constructed reference samples, where 8 different prediction directions plus an additional DC prediction (corresponding to mode 2, not shown in Figure 1) can be selected by the encoder. Note that the choice of the prediction type (4×4, 8×8, or 16×16) is also up to the encoder, subject to certain constraints on the choice of the corresponding luma transform size, as will be explained in Section II.B.

A distinguishing element of the new 8×8 intra prediction is given by a pre-filtering process of the reference samples (denoted by A–X, and Z in Figure 1). This lowpass filtering is conducted prior to the prediction step by applying a simple second-order binomial filter to the constructed luma reference samples.

B. *8×8 Transform*

The design of the first version of H.264/MPEG4-AVC includes two different 4×4 transforms for coding of the luma prediction error signal. A 4×4 integer transform is applied to all luma samples with an additional transform stage using the 4×4 Hadamard transform for all resulting 16 DC coefficients of a transformed MB in the case of a preceding 16×16 luma intra prediction only [1][4].

Besides the important property of low computational complexity, the use of small block-size transforms in H.264/MPEG4-AVC has the advantage of significantly reducing ringing artifacts. For high-fidelity video, however, the preservation of fine details and textures which generally requires larger basis functions becomes equally important. For achieving a proper trade-off between these conflicting objectives, the FRExt amendment includes an 8×8 integer transform and allows the encoder to choose adaptively between the 4×4 and 8×8 transform for luma samples on a MB level [6]. This capability is a simplified version of the concept of variable-block size transforms which was already under consideration during the first standardization phase of H.264/MPEG4-AVC [7][8].

The 2-D 8×8 transform in the FRExt amendment is specified in a separable way as a 1-D horizontal (row) transform followed by a 1-D vertical (column) transform, where the corresponding 1-D transform is given by the (non-normalized) transformation matrix $T_{8\times8}$ as follows:

$$T_{8x8} = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 12 & 10 & 6 & 3 & -3 & -6 & -10 & -12 \\ 8 & 4 & -4 & -8 & -8 & -4 & 4 & 8 \\ 10 & -3 & -12 & -6 & 6 & 12 & 3 & -10 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -12 & 3 & 10 & -10 & -3 & 12 & -6 \\ 4 & -8 & 8 & -4 & -4 & 8 & -8 & 4 \\ 3 & -6 & 10 & -12 & 12 & -10 & 6 & -3 \end{bmatrix}$$

Interestingly, both the forward and inverse 8×8 transform can be efficiently implemented via fast butterfly operations such that only shift and add operations in (8+*b*)-bit arithmetic precision for *b*-bit input video are required [6]. In fact, Table 1 indicates that the computational complexity for the butterfly implementation of the 8×8 inverse transform is only slightly higher than that of the corresponding four 4×4 inverse transforms.

TABLE 1: NUMBER OF OPERATIONS (# OPS) REQUIRED FOR THE 2-D 4×4 AND 8×8 INVERSE TRANSFORM IN H.264/MPEG4-AVC FREXT.

Transform size	Operation	# Ops per 8×8 block
4×4	+	64
	>>	16
8×8	+	64
	>>	20

By taking into account the six different squared norm values of the underlying 2-D basis functions, the processes of scaling, quantization, and scanning of 8×8 transform coefficients are specified as straightforward extensions of the corresponding operations already defined for the 4×4 transform in H.264/MPEG4-AVC. Moreover, CABAC is extended by employing 3 additional sets of context models for coding of 8×8 transform levels, whereas CAVLC is applied by regrouping the 8×8 transform levels into four groups of 4×4 levels (for more details, refer to [5]).

As already noted, the encoder can choose for each MB between the 4×4 and 8×8 transform and signal its choice by means of a flag. However, there are two restrictions imposed on the transform size selection process:

- For inter-coded MBs with one or more sub-partitions smaller than 8×8 (i.e., MC blocks of size 4x8, 8x4, or 4x4), the 4×4 luma transform is mandatory
- For intra-coded MBs, the 8×8 luma transform is chosen if and only if 8×8 luma spatial prediction is used

C. *Further Extensions*

In addition to support for extended sample bit depth and monochrome as well as 4:2:2 and 4:4:4 chroma formats, the FRExt amendment contains three other important tools:

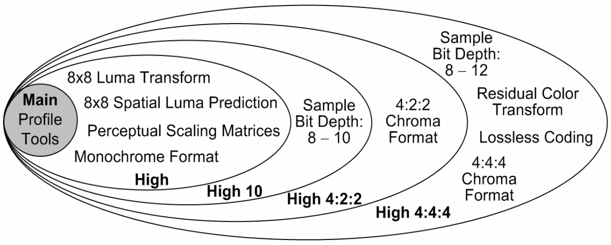

Figure 2: Illustration of the H.264/MPEG4-AVC FRExt profiles.

1. Encoder-specified *scaling matrices* for perceptual tuned, frequency-dependent quantization
2. A *residual color transform* consisting in a reversible integer-based color conversion from (4:4:4) RGB to the YCgCo color space applied to residual data only
3. A *lossless coding* capability requiring only a relatively simple bypass of transform and quantization

Furthermore, a few new supplemental enhancement information (SEI) messages for enabling enhancements of decoded video have been added to the FRExt amendment [5].

III. FRExt profiles and Application areas

As illustrated in Figure 2, the H.264/MPEG4-AVC FRExt amendment specifies a set of four new profiles which are constructed as nested subsets of capabilities. All four profiles inherit the tool set of the Main profile, and as their common intersection, the *High profile* (HP) additionally contains all major new tools for improving the coding efficiency. Relative to the Main profile (MP), these tools imply only moderate (if any) increases in complexity both in terms of implementation and computational costs (at the decoder side), as briefly discussed for the case of the 8×8 transform in the previous section. Therefore, the High profile with its restriction to 8-bit video in 4:2:0 chroma format (or optionally in monochrome format) is likely to replace the Main profile for prospective applications of H.264/MPEG4-AVC in typical consumer applications.

An indication of this trend is given by the fact that the H.264/MPEG4-AVC High profile has recently been included in some important application standards or specifications of industry consortia:

- The specifications TS 101 154 and TS 102 005 of DVB (Digital Video Broadcasting) specify HP for satellite, cable, and terrestrial broadcasting using MPEG-2 transport stream or IP, respectively
- The HD-DVD specification of the DVD Forum makes HP mandatory for every decoder
- The BD specification of the Blu-Ray Disc Association (BDA) also makes HP mandatory for every decoder

IV. Experimental Results

For a performance evaluation of the HP-specific 8×8 coding tools, as presented in Section II.A and II.B, a set of 7 progressive HD sequences (both at 1280x720@60Hz and 1920x1080@24Hz) with different characteristics has been used. The coding simulations were performed by using the H.264/MPEG4-AVC reference software version JM 9.4 and an MPEG-2 Main profile (MP@HL) conforming encoder, where both encoders have been controlled by using the same rate-distortion (RD) optimized Lagrangian strategy [3]. An I-frame refresh was performed every 500 ms, and two non-referenced B-pictures have been inserted between each two successive P-pictures. Full search motion estimation was performed with a search range of ±32 integer pixels. For H.264/MPEG4-AVC up to 3 reference frames were used.

Table 2: Average bit-rate savings over the whole test set for H.264/MPEG4-AVC HP using CABAC entropy coding in comparison with HP using CAVLC, MP using CABAC, and MPEG-2.

H.264/MPEG4-AVC HP using CAVLC	**H.264/MPEG4-AVC MP using CABAC**	**MPEG-2 MP@HL**
17.9%	9.9%	58.8%

Figure 3 (bottom row) shows the RD curves for the 'Raven' and the 'Book' sequences comparing H.264/MPEG4-AVC HP, MP (both with CABAC and CAVLC), and MPEG-2. Since these sequences are characterized by a predominant highly textured content, relatively large gains can be obtained in favor of HP due to the better frequency selectivity of the 8×8 luma transform. Averaged over the whole HD test set, HP achieves bit-rate savings of about 10% relative to MP (both using CABAC), as shown in Table 2. If, however, the 8×8 tool set of HP is used in conjunction with CAVLC, an average loss of about 18% is observed relative to HP using CABAC, which means that the CAVLC-driven HP leads, on average, to an objectively lower performance than that measured for the CABAC-driven MP (see Table 2). In comparison to MPEG-2, average bit-rate savings of about 59% were observed for the H.264/MPEG4-AVC High profile coder (using CABAC).

In another experiment, the HP-specific support for monochrome input was used in order to investigate the intra coding performance of H.264/MPEG4-AVC HP in comparison with JPEG2000 (Part 1) [9] for a set of 8 photographic monochrome test images (with resolutions from 512×512 up to 2048×3072 samples). Note that these images have been used extensively in the JPEG2000/JPEG-LS standardization work, whereas the H.264/MPEG4-AVC encoder was never before applied to that input data.

For this intra-only coding experiment, the Kakadu implementation [10] (version 2.2) of JPEG2000 was used as a reference, where the Kakadu encoder was driven in default lossy mode using an RD optimization for each given target rate point. The rate points were chosen to cover a range of 0.25 – 2 bits per pixel (bpp). Thus, in order to achieve the maximum compression performance for JPEG2000 and to guarantee a fair comparison, the SNR scalability feature of JPEG2000 was not utilized for our tests.

In Figure 3 (top right), two sets of RD curves are plotted – one for 'Lena' and another for the 'Barbara' image. As can be seen from the graph, the overall objective performance of JPEG2000 and H.264/MPEG4-AVC HP is nearly identical in both test cases. By disabling the 8×8 tool set in the HP encoder, a significant loss in PSNR of about 1 dB is observed for 'Barbara' with its well-known high-frequency

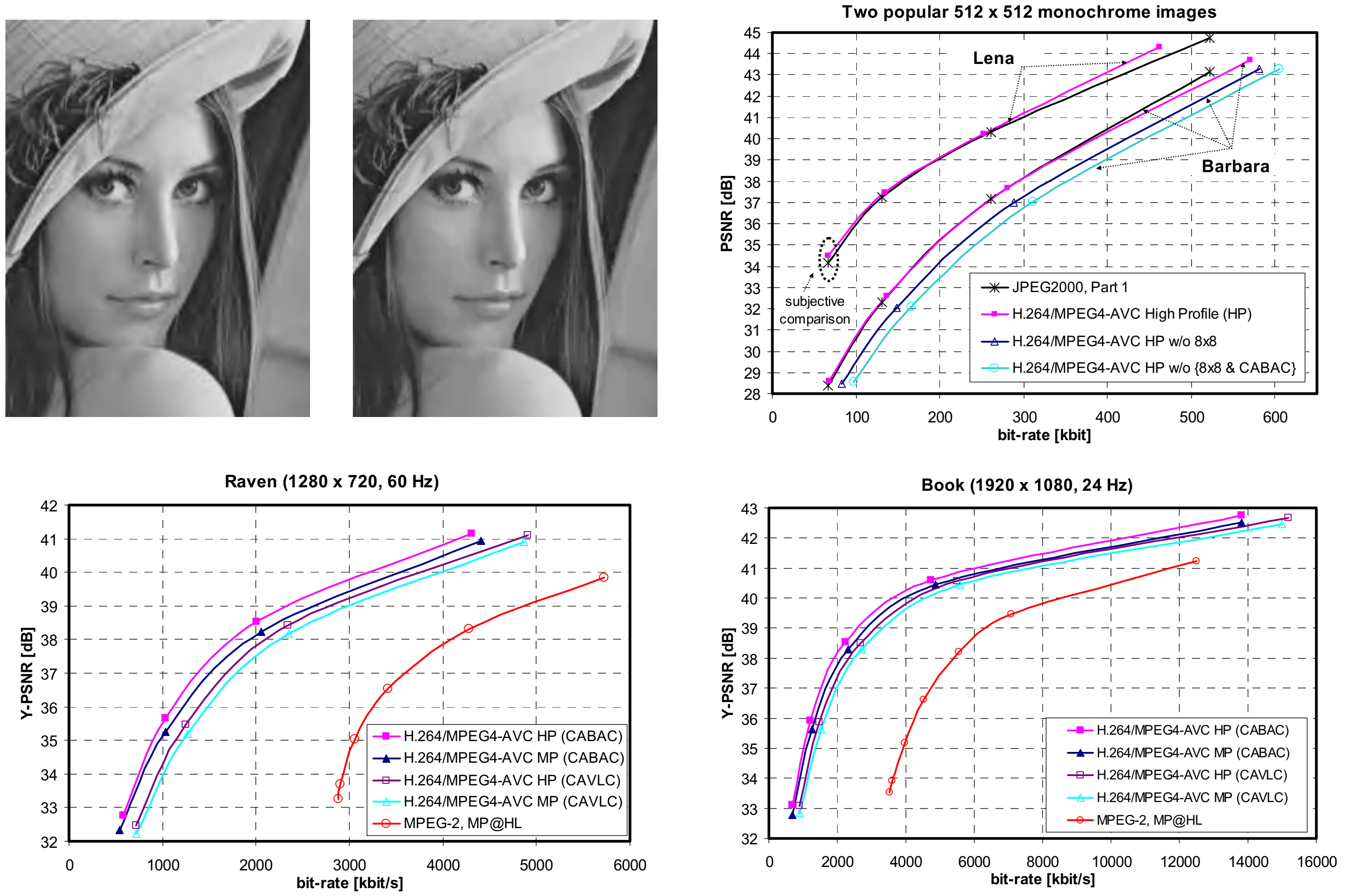

Figure 3: Top row: Detail of 'Lena' image comparing JPEG2000 (left) and H.264/MPEG4-AVC HP (middle) at a compression ratio of about 32:1 (0.25 bpp). The corresponding RD points are given as the lowest points on the curves in the RD plot (right) showing the RD behavior for both 'Lena' and 'Barbara'. Bottom row: RD graphs for the 'Raven' and the 'Book' sequence comparing H.264/MPEG4-AVC HP, MP (both using CABAC and CAVLC), and MPEG-2 MP@ML.

characteristics. Furthermore, going from CABAC to CAVLC entropy coding, results in an additional loss of 0.5 dB PSNR for that particular image, as shown Figure 3 (top right).

By averaging the intra coding results over the entire test image set and over all bit-rates, a gain of 0.5 dB PSNR was observed in favor of the H.264/MPEG4-AVC HP coder. In all cases, the subjective quality of the H.264/MPEG4-AVC HP encoded reconstruction was at least comparable to that of the corresponding JPEG2000 reconstruction at the same bit-rate. Figure 3 (top left) contains decoded samples of the 'Lena' image at 0.25 bpp for a comparison of the visual quality.

V. Conclusions

The H.264/MPEG4-AVC FRExt amendment has been presented with a particular focus on the High profile (HP) and its related coding tools. The High profile-specific tools offer a good trade-off between coding efficiency and implementation cost, especially when compared to the H.264/MPEG4-AVC Main profile (MP). Average bit-rate savings for the H.264/MPEG4-AVC High profile of 10% relative to MP (both using CABAC) and nearly 60% relative to MPEG-2 have been observed for a set of HD sequences. When restricted to intra-only coding, H.264/MPEG4-AVC HP outperforms the state-of-the-art in still-image coding represented by JPEG2000 on a set of monochrome test images by 0.5 dB average PSNR.

References

[1] ITU-T Recommendation H.264 & ISO/IEC 14496-10 (MPEG-4) AVC, "Advanced Video Coding for Generic Audiovisual Services", (version 1: 2003, versions 2: 2004) version 3: 2005.

[2] ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information – Part 2: Video," ITU-T Recommendation H.262 – ISO/IEC 13818-2 (MPEG-2), Nov. 1994.

[3] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," IEEE Trans. CSVT, vol. 13 (7), pp. 688-703, July 2003.

[4] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra,"Overview of the H.264/AVC video coding standard," IEEE Trans. CSVT, vol. 13 (7), pp. 560-576, July 2003.

[5] Joint Video Team of ITU-T and ISO/IEC: "Draft Text of H.264/AVC Fidelity Range Extensions Amendment", Doc. JVT-L047, Sept. 2004.

[6] S. Gordon, D. Marpe, and T. Wiegand, "Simplified Use of 8x8 Transforms", JVT Doc. JVT-K028, Munich, March 2004.

[7] M. Wien, "Variable Block-Size Transforms for H.264/AVC", IEEE Trans. CSVT, vol. 13 (7), pp. 604-613, July 2003.

[8] F. Bossen, "ABT Cleanup and Complexity Reduction", JVT Doc. JVT-E087, Geneva, Oct. 2002.

[9] ITU-T Rec. T.800 and ISO/IEC 15444-1, "JPEG2000 Image Coding System: Core Coding System" (JPEG2000 Part 1), 2000.

[10] Kakadu software available online: http://www.kakadusoftware.com.