Comparison of the Coding Efficiency of Video Coding Standards – Including High Efficiency Video Coding (HEVC)

Jens-Rainer Ohm, *Member, IEEE*, Gary J. Sullivan, *Fellow, IEEE*, Heiko Schwarz,
Thiow Keng Tan, *Senior Member, IEEE,* and Thomas Wiegand, *Fellow, IEEE*

***Abstract*—The compression capability of several generations of video coding standards is compared by means of PSNR and subjective testing results. A unified approach is applied to the analysis of designs including H.262/MPEG-2 Video, H.263, MPEG-4 Visual, H.264/MPEG-4 AVC, and HEVC. The results of subjective tests for WVGA and HD sequences indicate that HEVC encoders can achieve equivalent subjective reproduction quality as encoders that conform to H.264/MPEG-4 AVC when using approximately 50% less bit rate on average. The HEVC design is shown to be especially effective for low bit rates, high-resolution video content, and low-delay communication applications. The measured subjective improvement somewhat exceeds the improvement measured by the PSNR metric.**

***Index Terms*—Video compression, standards, HEVC, JCT-VC, MPEG, VCEG, H.264, MPEG-4, AVC.**

I. INTRODUCTION

THE primary goal of most digital video coding standards has been to optimize *coding efficiency*. Coding efficiency is the ability to minimize the bit rate necessary for representation of video content to reach a given level of video quality – or, as alternatively formulated, to maximize the video quality achievable within a given available bit rate.

The goal of this paper is to analyze the coding efficiency that can be achieved by use of the emerging high-efficiency video coding (HEVC) standard [1][2][3][4], relative to the coding efficiency characteristics of its major predecessors including H.262/MPEG-2 Video [5][6][7], H.263 [8], MPEG-4 Visual [9], and H.264/MPEG-4 AVC [10][11][12].

When designing a video coding standard for broad use, the standard is designed in order to give the developers of encoders and decoders as much freedom as possible to customize their implementations. This freedom is essential to enable a standard to be adapted to a wide variety of platform architectures, application environments, and computing resource constraints. This freedom is constrained by the need to achieve *interoperability* – i.e., to ensure that a video signal encoded by each vendor's products can be reliably decoded by others. This is ordinarily achieved by limiting the scope of the standard to two areas (cp. Fig. 1 in [11]):

1) Specifying the format of the data to be produced by a conforming encoder and constraining some characteristics of that data (such as its maximum bit rate and maximum frame rate), without specifying any aspects of how an encoder would process input video to produce the encoded data (leaving all pre-processing and algorithmic decision-making processes outside the scope of the standard), and
2) Specifying (or bounding the approximation of) the decoded results to be produced by a conforming decoder in response to a complete and error-free input from a conforming encoder, prior to any further operations to be performed on the decoded video (providing substantial freedom over the internal processing steps of the decoding process and leaving all post-processing, loss/error recovery, and display processing outside the scope as well).

This intentional limitation of scope complicates the analysis of coding efficiency for video coding standards, as most of the elements that affect the end-to-end quality characteristics are outside the scope of the standard. In this work, the emerging HEVC design is analyzed using a systematic approach that is largely similar in spirit to that previously applied to analysis of the first version of H.264/MPEG-4 AVC in [13]. A major emphasis in this analysis is the application of a disciplined and uniform approach for optimization of each of the video encoders. Additionally, a greater emphasis is placed on *subjective* video quality analysis than what was applied in [13], as the most important measure of video quality is the subjective perception of quality as experienced by human observers.

The paper is organized as follows: Section II briefly describes the syntax features of the investigated video coding standards and highlights the main coding tools that contribute to the coding efficiency improvement from one standard generation to the next. The uniform encoding approach that is used for all standards discussed in this paper is described in section III. In section IV, the current performance of the

Original manuscript received May 7, 2012, revised version received August 22, 2012.

J.-R. Ohm is with RWTH Aachen University, Aachen, Germany (e-mail: ohm@ient.rwth-aachen.de).

G. J. Sullivan is with Microsoft Corporation, Redmond, WA, 98052, USA (e-mail: garys@ieee.org).

H. Schwarz is with the Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute, Berlin, Germany (e-mail: heiko.schwarz@hhi.fraunhofer.de).

T. K. Tan is with M-Sphere Consulting Pte. Ltd., 808379, Singapore. He is a consultant for NTT DOCOMO, Inc. (e-mail: ttk@pacific.net.sg).

T. Wiegand is jointly affiliated with the Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute and the Berlin Institute of Technology, both in Berlin, Germany (e-mail: twiegand@ieee.org)

HEVC reference implementation is investigated in terms of tool-wise analysis, and in comparison to previous standards, as assessed by objective quality measurement (particularly PSNR). Section V provides results of the subjective quality testing of HEVC in comparison to the previous best-performing standard, H.264/MPEG-4 AVC.

II. SYNTAX OVERVIEW

The basic design of all major video coding standards since H.261 (in 1990) [14] follows the so-called block-based hybrid video coding approach. Each block of a picture is either *intra-picture* coded (a.k.a. coded in an "intra" coding mode), without referring to other pictures of the video sequence, or it is temporally predicted (i.e. *inter-picture* coded, a.k.a. coded in an "inter" coding mode), where the prediction signal is formed by a displaced block of an already coded picture. The latter technique is also referred to as motion-compensated prediction and represents the key concept for utilizing the large amount of temporal redundancy in video sequences. The prediction error signal (or the complete intra coded block) is processed using transform coding for exploiting spatial redundancy. The transform coefficients that are obtained by applying a decorrelating (linear or approximately linear) transform to the input signal are quantized and then entropy coded together with side information such as coding modes and motion parameters. Although all considered standards follow the same basic design, they differ in various aspects, which finally results in a significantly improved coding efficiency from one generation of standard to the next. In the following, we provide an overview of the main syntax features for the considered standards. The description is limited to coding tools for progressive-scan video that are relevant for the comparison in this paper. For further details, the reader is referred to the draft HEVC standard [4], the prior standards [5][8][9][10], and corresponding books [6][7][15] and overview articles [3][11][12].

In order to specify conformance points facilitating interoperability for different application areas, each standard defines particular *profiles*. A profile specifies a set of coding tools that can be employed in generating conforming bitstreams. We concentrate on the profiles that provide the best coding efficiency for progressive-scanned 8-bit-per-sample video with the 4:2:0 chroma sampling format, as the encoding of interlaced-scan video, high bit depths, and non-4:2:0 material is not in the central focus of the HEVC project for developing the first version of the standard.

A. ITU-T Rec. H.262 | ISO/IEC 13818-2 (MPEG-2 Video)

H.262/MPEG-2 Video [5] was developed as an official joint project of ITU-T and ISO/IEC JTC 1. It was finalized in 1994 and is still widely used for digital television and the DVD-video optical disc format. Similarly as for its predecessors H.261 [14] and MPEG-1 Video [16], each picture of a video sequence is partitioned into *macroblocks*, which consist of a 16×16 luma block and, in the 4:2:0 chroma sampling format, two associated 8×8 chroma blocks. The standard defines three picture types: I, P, and B pictures. I and P pictures are always coded in display/output order. In I pictures, all macroblocks are coded in intra coding mode, without referencing other pictures in the video sequence. A macroblock (MB) in a P picture can be either transmitted in intra or in inter mode. For the inter mode, the last previously coded I or P picture is used as reference picture. The displacement of an inter MB in a P picture relative to the reference picture is specified by a half-sample precision motion vector. The prediction signal at half-sample locations is obtained by bi-linear interpolation. In general, the motion vector is differentially coded using the motion vector of the MB to the left as a predictor. The standard includes syntax features that allow a particularly efficient signaling of zero-valued motion vectors. In H.262/MPEG-2 Video, B pictures have the property that they are coded after, but displayed before the previously coded I or P picture. For a B picture, two reference pictures can be employed: the I/P picture that precedes the B picture in display order and the I/P picture that succeeds it. When only one motion vector is used for motion compensation of a MB, the chosen reference picture is indicated by the coding mode. B pictures also provide an additional coding mode, for which the prediction signal is obtained by averaging prediction signals from both reference pictures. For this mode, which is referred to as the bi-prediction or bi-directional prediction mode, two motion vectors are transmitted. Consecutive runs of inter MBs in B pictures that use the same motion parameters as the MB to their left and do not include a prediction error signal can be indicated by a particularly efficient syntax.

For transform coding of intra MBs and the prediction errors of inter MBs, a DCT is applied to blocks of 8×8 samples. The DCT coefficients are represented using a scalar quantizer. For intra MBs, the reconstruction values are uniformly distributed, while for inter MBs, the distance between zero and the first non-zero reconstruction values is increased to three halves of the quantization step size. The intra DC coefficients are differentially coded using the intra DC coefficient of the block to their left (if available) as their predicted value. For perceptual optimization, the standard supports the usage of quantization weighting matrices, by which effectively different quantization step sizes can be used for different transform coefficient frequencies. The transform coefficients of a block are scanned in a zig-zag manner and transmitted using two-dimensional run-level variable-length coding (VLC). Two VLC tables are specified for quantized transform coefficients (a.k.a. "levels"). One table is used for inter MBs. For intra MBs, the employed table can be selected at the picture level.

The most widely implemented profile of H.262/MPEG-2 Video is the Main Profile. It supports video coding with the 4:2:0 chroma sampling format and includes all tools that significantly contribute to coding efficiency. The Main Profile is used for the comparisons in this paper.

B. ITU-T Recommendation H.263

The first version of ITU-T Rec. H.263 [8] defines syntax features that are very similar to those of H.262/MPEG-2 Video, but it includes some changes that make it more efficient for low-delay low bit rate coding. The coding of motion vectors has been improved by using the component-wise median of the motion vectors of three neighboring previously decoded blocks as the motion vector predictor. The transform coefficient levels are coded using a three-dimensional run-level-last VLC, with tables optimized for lower bit rates. The first version of H.263 contains four annexes (annexes D through G) that specify additional coding options, among which annexes D and F are frequently used for improving coding efficiency. The usage of annex D allows motion vectors to point outside the reference picture, a key feature that is not permitted in H.262/MPEG-2 Video. Annex F introduces a coding mode for P pictures, the inter 8×8 mode, in which four motion vectors are transmitted for a MB, each for an 8×8 sub-block. It further specifies the usage of overlapped block motion compensation.

The second and third versions of H.263, which are often called H.263+ and H.263++, respectively, add several optional coding features in the form of annexes. Annex I improves the intra coding by supporting a prediction of intra AC coefficients, defining alternative scan patterns for horizontally and vertically predicted blocks, and adding a specialized quantization and VLC for intra coefficients. Annex J specifies a deblocking filter that is applied inside the motion compensation loop. Annex O adds scalability support, which includes a specification of B pictures roughly similar to those in H.262/MPEG-2 Video. Some limitations of version 1 in terms of quantization are removed by annex T, which also improves the chroma fidelity by specifying a smaller quantization step size for chroma coefficients than for luma coefficients. Annex U introduces the concept of multiple reference pictures. With this feature, motion-compensated prediction is not restricted to use just the last decoded I/P picture (or, for coded B pictures using annex O, the last two I/P pictures) as a reference picture. Instead, multiple decoded reference pictures are inserted into a picture buffer and can be used for inter prediction. For each motion vector, a reference picture index is transmitted, which indicates the employed reference picture for the corresponding block. The other annexes in H.263+ and H.263++ mainly provide additional functionalities such as the specification of features for improved error resilience.

The H.263 profiles that provide the best coding efficiency are the Conversational High Compression (CHC) profile and the High Latency Profile (HLP). The CHC profile includes most of the optional features (annexes D, F, I, J, T, and U) that provide enhanced coding efficiency for low-delay applications. The High Latency Profile adds the support of B pictures (as defined in annex O) to the coding efficiency tools of the CHC profile and is targeted for applications that allow a higher coding delay.

C. *ISO/IEC 14496-2 (MPEG-4 Visual)*

MPEG-4 Visual [9], a.k.a. Part 2 of the MPEG-4 suite is backward-compatible to H.263 in the sense that each conforming MPEG-4 decoder must be capable of decoding H.263 Baseline bitstreams (i.e. bitstreams that use no H.263 optional annex features). Similarly as for annex F in H.263, the inter prediction in MPEG-4 can be done with 16×16 or 8×8 blocks. While the first version of MPEG-4 only supports motion compensation with half-sample precision motion vectors and bi-linear interpolation (similar to H.262/MPEG-2 Video and H.263), version 2 added support for quarter-sample precision motion vectors. The luma prediction signal at half-sample locations is generated using an 8-tap interpolation filter. For generating the quarter-sample positions, bi-linear interpolation of the integer- and half-sample positions is used. The chroma prediction signal is generated by bi-linear interpolation. Motion vectors are differentially coded using a component-wise median prediction and are allowed to point outside the reference picture. MPEG-4 Visual supports B pictures (in some profiles), but it does not support the feature of multiple reference pictures (except on a slice basis for loss resilience purposes) and it does not specify a deblocking filter inside the motion compensation loop.

The transform coding in MPEG-4 is basically similar to that of H.262/MPEG-2 Video and H.263. However, two different quantization methods are supported. The first quantization method, which is sometimes referred to as MPEG-style quantization, supports quantization weighting matrices similarly to H.262/MPEG-2 Video. With the second quantization method, which is called H.263-style quantization, the same quantization step size is used for all transform coefficients with the exception of the DC coefficient in intra blocks. The transform coefficient levels are coded using a three-dimensional run-level-last code as in H.263. Similarly as in annex I of H.263, MPEG-4 Visual also supports the prediction of AC coefficients in intra blocks as well as alternative scan patterns for horizontally and vertically predicted intra blocks and the usage of a separate VLC table for intra coefficients.

For the comparisons in this paper, we used the Advanced Simple Profile (ASP) of MPEG-4 Visual, which includes all relevant coding tools. We generally enabled quarter-sample precision motion vectors. MPEG-4 ASP additionally includes global motion compensation. Due to the limited benefits experienced in practice and the complexity and general difficulty of estimating global motion fields suitable for improving the coding efficiency, this feature is rarely supported in encoder implementations and is also not used in our comparison.

D. *ITU-T Rec. H.264 | ISO/IEC 14496-10 (MPEG-4 AVC)*

H.264/MPEG-4 AVC [10][12] is the second video coding standard that was jointly developed by ITU-T VCEG and ISO/IEC MPEG. It still uses the concept of 16×16 macroblocks, but contains many additional features. One of the most obvious differences from older standards is its increased flexibility for inter coding. For the purpose of motion-compensated prediction, a macroblock can be partitioned into square and rectangular block shapes with sizes ranging from 4×4 to 16×16 luma samples. H.264/MPEG-4 AVC also supports multiple reference pictures. Similarly to annex U of H.263, motion vectors are associated with a reference picture index for specifying the employed reference picture. The motion vectors are transmitted using quarter-sample precision relative to the luma sampling grid. Luma prediction values at half-sample locations are generated using a 6-tap interpolation filter and prediction values at quarter-sample locations are obtained by averaging two values at integer- and half-sample positions. Weighted prediction can be applied using a scaling and offset of the prediction signal. For the chroma components, a bi-linear interpolation is applied. In general, motion vectors are predicted by the component-wise median of the motion vectors of three neighboring previously decoded blocks. For 16×8 and 8×16 blocks, the predictor is given by the motion vector of a single already decoded neighboring block, where the chosen neighboring block depends on the location of the block inside a macroblock. In contrast to prior coding standards, the concept of B pictures is generalized and the picture coding type is decoupled from the coding order and the usage as a reference picture. Instead of I, P, and B pictures, the standard actually specifies I, P, and B slices. A picture can contain slices of different types and a picture can be used as a reference for inter prediction of subsequent pictures independently of its slice coding types. This generalization allowed the usage of prediction structures such as hierarchical B pictures [17] that show improved coding efficiency compared to the IBBP coding typically used for H.262/MPEG-2 Video.

H.264/MPEG-4 AVC also includes a modified design for intra coding. While in previous standards some of the DCT coefficients can be predicted from neighboring intra blocks, the intra prediction in H.264/MPEG-4 AVC is done in the spatial domain by referring to neighboring samples of previously decoded blocks. The luma signal of a macroblock can be either predicted as a single 16×16 block or it can be partitioned into 4×4 or 8×8 blocks with each block being predicted separately. For 4×4 and 8×8 blocks, nine prediction modes specifying different prediction directions are supported. In the intra 16×16 mode and for the chroma components, four intra prediction modes are specified.

For transform coding, H.264/MPEG-4 AVC specifies a 4×4 and an 8×8 transform. While chroma blocks are always coded using the 4×4 transform, the transform size for the luma component can be selected on a macroblock basis. For intra MBs, the transform size is coupled to the employed intra prediction block size. An additional 2×2 Hadamard transform is applied to the four DC coefficients of each chroma component. For the intra 16×16 mode, a similar second-level Hadamard transform is also applied to the 4×4 DC coefficients of the luma signal. In contrast to previous standards, the inverse transforms are specified by exact integer operations, so that, in error-free environments, the reconstructed pictures in the encoder and decoder are always exactly the same. The transform coefficients are represented using a uniform reconstruction quantizer, i.e., without the extra-wide dead-zone that is found in older standards. Similar to H.262/MPEG-2 Video and MPEG-4 Visual, H.264/MPEG-4 AVC also supports the usage of quantization weighting matrices. The transform coefficient levels of a block are generally scanned in a zig-zag fashion.

For entropy coding of all macroblock syntax elements, H.264/MPEG-4 AVC specifies two methods. The first entropy coding method, which is known as context-adaptive variable-length coding (CAVLC), uses a single codeword set for all syntax elements except the transform coefficient levels. The approach for coding the transform coefficients basically uses the concept of run-level coding as in prior standards. However, the efficiency is improved by switching between VLC tables depending on the values of previously transmitted syntax elements. The second entropy coding method specifies context-adaptive binary arithmetic coding (CABAC) by which the coding efficiency is improved relative to CAVLC. The statistics of previously coded symbols are used for estimating conditional probabilities for binary symbols, which are transmitted using arithmetic coding. Inter-symbol dependencies are exploited by switching between several estimated probability models based on previously decoded symbols in neighboring blocks. Similar to annex J of H.263, H.264/MPEG-4 AVC includes a deblocking filter inside the motion compensation loop. The strength of the filtering is adaptively controlled by the values of several syntax elements.

The High Profile of H.264/MPEG-4 AVC includes all tools that contribute to the coding efficiency for 8-bit-per-sample video in 4:2:0 format, and is used for the comparison in this paper. Because of its limited benefit for typical video test sequences and the difficulty of optimizing its parameters, the weighted prediction feature is not applied in the testing.

E. HEVC (draft 8 of July 2012)

High Efficiency Video Coding (HEVC) [4] is the name of the current joint standardization project of ITU-T VCEG and ISO/IEC MPEG, currently under development in a collaboration known as the Joint Collaborative Team on Video Coding (JCT-VC). It is planned to finalize the standard in early 2013. In the following, a brief overview of the main changes relative to H.264/MPEG-4 AVC is provided. For a more detailed description, the reader is referred to the overview in [2].

In HEVC, a picture is partitioned into coding tree blocks (CTBs). The size of the CTBs can be chosen by the encoder according to its architectural characteristics and the needs of its application environment, which may impose limitations such as encoder/decoder delay constraints and memory requirements. A luma CTB covers a rectangular picture area of $N\times N$ samples of the luma component and the corresponding chroma CTBs cover each $(N/2)\times(N/2)$ samples of each of the two chroma components. The value of N is signaled inside the bitstream, and can be 16, 32, or 64. The luma CTB and the two chroma CTBs, together with the associated syntax, form a coding tree unit (CTU). The CTU is the basic processing unit of the standard to specify the decoding process (conceptually corresponding to a macroblock in prior standards).

The blocks specified as luma and chroma CTBs can be further partitioned into multiple coding blocks (CBs). The CTU contains a quadtree syntax that allows for splitting into blocks of variable size considering the characteristics of the region that is covered by the CTB. The size of the CB can range from the same size as the CTB to a minimum size (8×8 luma samples or larger) that is specified by a syntax element conveyed to the decoder. The luma CB and the chroma CBs, together with the associated syntax, form a coding unit (CU).

For each CU, a prediction mode is signaled, which can be either an intra or inter mode. When intra prediction is chosen, one of 35 spatial intra prediction modes is signaled for the luma CB. When the luma CB has the indicated smallest allowable size, it is also possible to signal one intra prediction mode for each of its four square sub-blocks. For both chroma CBs, a single intra prediction mode is selected. Except for some special cases, it specifies using horizontal, vertical, planar, or DC prediction, or using the same prediction mode that was used for luma. The intra prediction mode is applied separately for each transform block.

For inter coded CUs, the luma and chroma CBs correspond to one, two, or four luma and chroma PBs. The smallest luma PB size is 4×8 or 8×4 samples. The luma and chroma PBs, together with the associated syntax, form a prediction unit (PU). Each PU contains one or two motion vectors for uni-predictive or bi-predictive coding, respectively. All PBs of a CB can have the same size, or, when asymmetric motion partitioning (AMP) is used, a luma CB of size $N \times N$ can also be split into two luma PBs, where one of the luma PBs covers $N \times (N/4)$ or $(N/4) \times N$ samples and the other luma PB covers the remaining $N \times (3 \cdot N/4)$ or $(3 \cdot N/4) \times N$ area of the CB. The AMP splitting is also applied to chroma CBs accordingly.

Similar to H.264/MPEG-4 AVC, HEVC supports quarter-sample precision motion vectors. The luma prediction signal for all fractional-sample locations is generated by separable 7- or 8-tap filters (depending on the sub-sample shift). For chroma, 4-tap interpolation filters are applied. HEVC also supports multiple reference pictures, and the concepts of I, P, and B slices are basically unchanged from H.264/MPEG-4 AVC. Weighted prediction is also supported in a similar manner.

The coding of motion parameters has been substantially improved compared to prior standards. HEVC supports a so-called merge mode, in which no motion parameters are coded. Instead, a candidate list of motion parameters is derived for the corresponding PU. In general, the candidate list includes the motion parameters of spatially neighboring blocks as well as temporally-predicted motion parameters that are derived based on the motion data of a co-located block in a reference picture. The chosen set of motion parameters is signaled by transmitting an index into the candidate list. The usage of large block sizes for motion compensation and the merge mode allow a very efficient signaling of motion data for large consistently displaced picture areas. If a PU is not coded using the merge mode, the associated reference indices and motion vector prediction differences are transmitted. Prediction is done using the advanced motion vector prediction (AMVP) algorithm. In AMVP, for each motion vector, a candidate list is constructed, which can include the motion vectors of neighboring blocks with the same reference index as well as a temporally predicted motion vector. The motion vector is coded by transmitting an index into the candidate list for specifying the chosen predictor and coding a difference vector.

For coding the inter or intra prediction residual signal of a luma CB, the CB is either represented as a single luma transform block (TB) or is split into four equal-size luma TBs. If the luma CB is split, each resulting luma TB can be further split into four smaller luma TBs. The same splitting applies to the chroma CB (except that 4×4 chroma TBs are not further split) and the scheme is called the residual quadtree (RQT), with the luma and chroma TBs and associated syntax forming a transform unit (TU). For each TU, the luma and chroma TBs are each transformed using a separable 2-D transform. Maximum and minimum TB sizes are selected by the encoder. All TBs are square with block sizes of 4×4, 8×8, 16×16, or 32×32. Similarly as in H.264/MPEG-4 AVC, the inverse transforms are specified by exact integer operations. In general, the transforms represent integer approximations of a DCT. For intra TUs of size 4×4, an alternative transform representing an approximation of a discrete sine transform (DST) is used.

All slice data syntax elements are entropy coded using CABAC, which is similar to the CABAC coding in H.264/MPEG-4 AVC. However, the coding of transform coefficient levels has been improved by using a more sophisticated context selection scheme, which is particularly efficient for larger transform sizes. Besides a deblocking filter, the HEVC design includes a sample-adaptive offset (SAO) operation inside the motion compensation loop. SAO classifies the reconstructed samples into different categories (e.g. depending on edge orientation) and reduces the distortion by adding a separate offset for each class of samples.

The current HEVC draft specifies a single profile: the Main Profile (MP). It includes all coding tools as described above, and supports the coding of 8-bit-per-sample video in 4:2:0 chroma format. For some comparisons in this paper, we used a modified configuration, where some coding tools are disabled. As for H.264/MPEG-4 AVC, the weighted prediction feature of HEVC has not been used for the simulations in this paper.

III. ENCODER CONTROL

Since all video coding standards of ITU-T and ISO/IEC JTC 1 specify only the bitstream syntax and the decoding process, they do not guarantee any particular coding efficiency. In this paper, all encoders are operated using the same encoding techniques, where the main focus of the comparison is on investigating the coding efficiency that is achievable by the bitstream syntax. Encoding constraints such as real-time operation or error robustness are not taken into account.

In order to keep the paper self-contained, we briefly review the Lagrangian encoding technique used in this paper. The task of an encoder control for a particular coding standard is to determine the values of the syntax elements, and thus the bitstream $\boldsymbol{b}$, for a given input sequence $\boldsymbol{s}$ in a way that the distortion $D(\boldsymbol{s}, \boldsymbol{s}')$ between the input sequence $\boldsymbol{s}$ and its reconstruction $\boldsymbol{s}' = \boldsymbol{s}'(\boldsymbol{b})$ is minimized subject to a set of constraints, which usually includes constraints for the average and maximum bit rate and the maximum coding delay. Let B_c be the set of all conforming bitstreams that obey the given set of constraints. For any particular distortion measure $D(\boldsymbol{s}, \boldsymbol{s}')$, the optimal bitstream in the rate–distortion sense is given by

$$\boldsymbol{b}^* = \underset{\boldsymbol{b} \in B_c}{\arg\min}\, D(\boldsymbol{s}, \boldsymbol{s}'(\boldsymbol{b})). \tag{1}$$

Due to the huge parameter space and encoding delay, it is impossible to directly apply the minimization in (1). Instead, the overall minimization problem is split into a series of smaller minimization problems by partly neglecting spatial and temporal interdependencies between coding decisions.

Let $\boldsymbol{s}_k$ be a set of source samples, such as a video picture or a block of a video picture, and let $\boldsymbol{p} \in P_k$ be a vector of coding decisions (or syntax element values) out of a set P_k of coding options for the set of source samples $\boldsymbol{s}_k$. The problem of finding the coding decisions $\boldsymbol{p}$ that minimize a distortion measure $D_k(\boldsymbol{p}) = D(\boldsymbol{s}_k, \boldsymbol{s}'_k)$ between the original samples $\boldsymbol{s}_k$ and their reconstructions $\boldsymbol{s}'_k = \boldsymbol{s}'_k(\boldsymbol{p})$ subject to a rate constraint R_c can be formulated as

$$\min_{\boldsymbol{p}\in P_k} D_k(\boldsymbol{p}) \qquad \text{subject to} \qquad R_k(\boldsymbol{p}) \le R_c, \tag{2}$$

where $R_k(\boldsymbol{p})$ represents the number of bits that are required for signaling the coding decisions $\boldsymbol{p}$ in the bitstream. Other constraints, such as the maximum coding delay or the minimum interval between random access points, can be considered by selecting appropriate prediction structures and coding options. The constrained minimization problem in (2) can be reformulated as an unconstrained minimization [13][18]–[23],

$$\min_{\boldsymbol{p}\in P_k} D_k(\boldsymbol{p}) + \lambda \cdot R_k(\boldsymbol{p}), \tag{3}$$

where $\lambda \ge 0$ denotes the so-called Lagrange multiplier.

If a set of source samples $\boldsymbol{s}_k$ can be partitioned into a number of subsets $\boldsymbol{s}_{k,i}$ in a way that the associated coding decisions $\boldsymbol{p}_i$ are independent of each other and an additive distortion measure $D_{k,i}(\boldsymbol{p}_i)$ is used, the minimization problem in (3) can be written as

$$\sum_i \min_{\boldsymbol{p}_i\in P_{k,i}} D_{k,i}(\boldsymbol{p}_i) + \lambda \cdot R_{k,i}(\boldsymbol{p}_i). \tag{4}$$

The optimal solution of (3) can be obtained by independently selecting the coding options $\boldsymbol{p}_i$ for the subsets $\boldsymbol{s}_{k,i}$. Although most coding decisions in a video encoder cannot be considered independent, for a practical applicability of the Lagrangian encoder control, it is required to split the overall optimization problem into a set of feasible decisions. While past decisions are taken into account by determining the distortion and rate terms based on already coded samples, the impact of a decision on future samples and coding decisions is ignored.

The concept of the described Lagrangian encoder control is applied for mode decision, motion estimation, and quantization. The used distortion measures D are defined as

$$\sum_{i\in B} |s_i - s_i'|^p, \tag{5}$$

with $p = 1$ for the sum of absolute differences (SAD) and $p = 2$ for the sum of squared differences (SSD). s_i and s_i' represent the original and reconstructed samples, respectively, of a considered block B. Except for motion estimation, we use SSD as the distortion measure for all coding decisions. Hence, all encoders are basically optimized with respect to the mean squared error (MSE) or peak signal-to-noise ratio (PSNR). The subjective quality of the reconstructed video, which is the ultimate video quality measure, is not directly taken into account during encoding. Nonetheless, this encoder control method usually also provides a good trade-off between subjective quality and bit rate.

A. Mode decision

The minimization of a Lagrangian cost function for mode decision was proposed in [21][22]. The investigated video coding standards provide different coding modes c for coding a block of samples $\boldsymbol{s}_k$, such as a macroblock or a coding unit. The coding modes may represent intra or inter prediction modes or partitions for motion-compensated prediction or transform coding. Given the set C_k of applicable coding modes for a block of samples $\boldsymbol{s}_k$, the used coding mode is chosen according to

$$c^* = \arg\min_{c\in C_k} D_k(c) + \lambda \cdot R_k(c), \tag{6}$$

where the distortion term $D_k(c)$ represents the SSD between the original block $\boldsymbol{s}_k$ and its reconstruction $\boldsymbol{s}_k'$ that is obtained by coding the block $\boldsymbol{s}_k$ with the mode c. The term $R_k(c)$ represents the number of bits (or an estimate thereof) that are required for representing the block $\boldsymbol{s}_k$ using the coding mode c for the given bitstream syntax. It includes the bits required for signaling the coding mode and the associated side information (e.g., motion vectors, reference indices, intra prediction modes, and coding modes for sub-blocks of $\boldsymbol{s}_k$) as well as the bits required for transmitting the transform coefficient levels representing the residual signal. A coding mode is often associated with additional parameters such as coding modes for sub-blocks, motion parameters, and transform coefficient levels. While coding modes for sub-blocks are determined in advance according to (6), motion parameters and transform coefficient levels are chosen as described in subsections III.B and III.C, respectively. For calculating the distortion and rate terms for the different coding modes, decisions for already coded blocks of samples are taken into account (e.g., by considering the correct predictors or context models).

For the investigated encoders, the described mode decision process is used for the following:

- the decision on whether a macroblock or a coding unit is coded using intra or inter prediction;
- the determination of intra prediction modes;
- the selection of a subdivision for a block or coding unit into sub-blocks for inter prediction;
- the selection of the transform size or transform subdivision for a macroblock or coding unit;
- the subdivision of a coding unit into smaller coding units for HEVC.

A similar process is also used for determining the SAO parameters in HEVC.

B. Motion estimation

The minimization of a Lagrangian cost function for motion estimation was proposed in [23]. Given a reference picture list R and a candidate set M of motion vectors, the motion parameters for a block $\boldsymbol{s}_k$, which consist of a displacement or motion vector $\boldsymbol{m} = [m_x, m_y]$ and, if applicable, a reference index r, are determined according to

$$(r^*, \boldsymbol{m}^*) = \arg\min_{r\in R, \boldsymbol{m}\in M} D_k(r, \boldsymbol{m}) + \lambda_M \cdot R_k(r, \boldsymbol{m}). \tag{7}$$

The rate term $R_k(r, \boldsymbol{m})$ represents an estimate of the number of bits that is required for transmitting the motion parameters. For determining the rate term, the motion vector predictor for the current block (or, for HEVC, one of the possible predictors) is taken into account.

For each candidate reference index r, the motion search first proceeds over a defined set of integer-sample precision displacement vectors. For this stage, the distortion $D_k(r, \boldsymbol{m})$ is measured as the sum of absolute differences (SAD) between the block $\boldsymbol{s}_k$ and the displaced reference block in the reference pictures indicated by the reference index r. For the integer-sample precision search, all encoders use the same fast motion estimation strategy (the one implemented in the HM-8.0 reference software [24]). Given the selected integer-sample precision displacement vector, the eight surrounding half-sample precision displacement vectors are evaluated. Then, for the coding standards supporting quarter-sample precision motion vectors, the half-sample refinement is followed by a quarter-sample refinement, in which the eight quarter-sample precision vectors that surround the selected half-sample precision motion vector are tested. The distortion measure that is used for the sub-sample refinements is the SAD in the Hadamard domain. The difference between the original block $\boldsymbol{s}_k$ and its motion-compensated prediction signal given by r and $\boldsymbol{m}$, is transformed using a block-wise 4×4 or 8×8 Hadamard transform, and the distortion is obtained by summing up the absolute transform coefficients. As has been experimentally found, the usage of the SAD in the Hadamard domain usually improves the coding efficiency in comparison to using the SAD in the sample domain [25]. Due to its computationally demanding calculation, the Hadamard-domain measurement is only used for the sub-sample refinement.

In HEVC, the motion vector predictor for a block is not fixed, but can be chosen out of a set of candidate predictors. The used predictor is determined by minimizing the number of bits required for coding the motion vector $\boldsymbol{m}$. Finally, given the selected motion vector for each reference index r, the used reference index is selected according to (7), where the SAD in the Hadamard domain is used as the distortion measure.

For bi-predictively coded blocks, two motion vectors and reference indices need to be determined. The initial motion parameters for each reference list are determined independently by minimizing the cost measure in (7). This is followed by an iterative refinement step [26], in which one motion vector is held constant and for the other motion vector, a refinement search is carried out. For this iterative refinement, the distortions are calculated based on the prediction signal that is obtained by bi-prediction. The decision whether a block is coded using a single or two motion vectors is also based on a Lagrangian function similar to (7), where the SAD in the Hadamard domain is used as distortion measure and the rate term includes all bits required for coding the motion parameters.

Due to the different distortion measure, the Lagrange multiplier λ_M that is used for determining the motion parameters is different from the Lagrange multiplier λ used in mode decision. In [20][27], the simple relationship $\lambda_M = \sqrt{\lambda}$ between those parameters is suggested, which is also used for the investigations in this paper.

C. Quantization

In classical scalar quantization, fixed thresholds are used for determining the quantization index of an input quantity. But since the syntax for transmitting the transform coefficient levels in image and video coding uses interdependencies between the transform coefficient levels of a block, the rate–distortion efficiency can be improved by taking into account the number of bits required for transmitting the transform coefficient levels. An approach for determining transform coefficient levels based on a minimization of a Lagrangian function has been proposed in [28] for H.262/MPEG-2 Video. In [29][30], similar concepts for a rate–distortion optimized quantization (RDOQ) are described for H.264/MPEG-4 AVC. The general idea is to select the vector of transform coefficient levels $\boldsymbol{l}$ for a transform block $\boldsymbol{t}$ by minimizing the function

$$\boldsymbol{l}^* = \underset{\boldsymbol{l} \in L^N}{\arg\min}\, D(\boldsymbol{l}) + \lambda \cdot R(\boldsymbol{l}), \tag{8}$$

where L^N represents the vector space of the N transform coefficient levels and $D(\boldsymbol{l})$ and $R(\boldsymbol{l})$ denote the distortion and the number of bits associated with the selection $\boldsymbol{l}$ for the considered transform block. As distortion measure, we use the SSD. Since the transforms specified in the investigated standards have orthogonal basis functions (if neglecting rounding effects), the SSD can be directly calculated in the transform domain, $D(\boldsymbol{l}) = \sum_i D(l_i)$. It is of course infeasible to proceed the minimization over the entire product space L^N. However, it is possible to apply a suitable decision process by which none or only some minor interdependencies are neglected. The actual quantization process is highly dependent on the bitstream syntax. As an example, we briefly describe the quantization for HEVC in the following.

In HEVC, a transform block is represented by a flag indicating whether the block contains non-zero transform coefficient levels, the location of the last non-zero level in scanning order, a flag for sub-blocks indicating whether the sub-block contains non-zero levels, and syntax elements for representing the actual levels. The quantization process basically consists of the following ordered steps:

1. For each scanning position i, the selected level l_i^* is determined assuming that the scanning position lies in a non-zero sub-block and i is less than or equal to the last scanning position. This decision is based on minimization of the function $D(l_i) + \lambda \cdot R_i(l_i)$, where $D(l_i)$ represents the (normalized) squared error for the considered transform coefficient and $R_i(l_i)$ denotes the number of bits that would be required for transmitting the level l_i. For reducing complexity, the set of tested levels can be limited, e.g., to the two levels that would be obtained by a mathematically correct rounding and a rounding toward zero of the original transform coefficient divided by the quantization step size.
2. For each sub-block, the rate–distortion cost for the determined levels is compared with the rate–distortion cost that is obtained when all levels of the sub-block are set to zero. If the latter cost is smaller, all levels of the sub-block are set to zero.
3. Finally, the flag indicating whether the block contains non-zero levels and the position of the last non-zero level are determined by calculating the rate–distortion cost that is obtained when all levels of the transform block are set equal to zero and the rate–distortion costs that are obtained when all levels that precede a particular non-zero level are set equal to zero. The setting that yields the minimum rate–distortion costs determines the chosen set of transform coefficient levels.

D. *Quantization parameters and Lagrange multipliers*

For all results presented in this paper, the quantization parameter QP and the Lagrange multiplier λ are held constant for all macroblocks or coding units of a video picture. The Lagrange multiplier is set according to

$$\lambda = \alpha \cdot Q^2, \tag{9}$$

where Q denotes the quantization step size, which is controlled by the quantization parameter QP (cp. [20][27]). Given the quantization parameter QP_I for intra pictures, the quantization parameters for all other pictures and the factors α are set using a deterministic approach. The actual chosen values depend on the used prediction structure and have been found in an experimental way.

IV. PERFORMANCE MEASURMENT OF THE HEVC REFERENCE CODEC IMPLEMENTATION

A. *Description of criteria*

The Bjøntegaard measurement method [31] for calculating objective differences between rate–distortion curves was used as evaluation criterion in this section. The average differences in bit rate between two curves, measured in percent, are reported here. In the original measurement method, separate rate–distortion curves for the luma and chroma components were used, hence resulting in three different average bit rate differences, one for each of the components. Separating these measurements is not ideal and sometimes confusing, as tradeoffs between the performance of the luma and chroma components are not taken into account.

In the used method, the rate–distortion curves of the combined luma and chroma components are used. The combined peak signal-to-noise ratio ($\mathrm{PSNR}_{\mathrm{YUV}}$) is first calculated as the weighted sum of the peak signal-to-noise ratio per picture of the individual components ($\mathrm{PSNR}_{\mathrm{Y}}$, $\mathrm{PSNR}_{\mathrm{U}}$ and $\mathrm{PSNR}_{\mathrm{V}}$),

$$\mathrm{PSNR}_{\mathrm{YUV}} = (6 \cdot \mathrm{PSNR}_{\mathrm{Y}} + \mathrm{PSNR}_{\mathrm{U}} + \mathrm{PSNR}_{\mathrm{V}}) / 8, \tag{10}$$

where $\mathrm{PSNR}_{\mathrm{Y}}$, $\mathrm{PSNR}_{\mathrm{U}}$, $\mathrm{PSNR}_{\mathrm{V}}$ are each computed as

$$\mathrm{PSNR} = 10 \cdot \log_{10}((2^B - 1)^2 / \mathrm{MSE}), \tag{11}$$

$B = 8$ is the number of bits per sample of the video signal to be coded and the MSE is the SSD divided by the number of samples in the signal. The PSNR measurements per video sequence are computed by averaging the per-picture measurements.

Using the bit rate and the combined $\mathrm{PSNR}_{\mathrm{YUV}}$ as the input to the Bjøntegaard measurement method gives a single average difference in bit rate that (at least partially) takes into account the tradeoffs between luma and chroma component fidelity.

B. *Results about the benefit of some representative tools*

In general, it is difficult to fairly assess the benefit of a video compression algorithm on a tool-by-tool basis, as the adequate design is reflected by an appropriate *combination* of tools. For example, introduction of larger block structures has impact on motion vector compression (particularly in the case of homogeneous motion), but should be accompanied by incorporation of larger transform structures as well. Therefore, the subsequent paragraphs are intended to give some idea about the benefits of some representative elements when switched on in the HEVC design, compared to a configuration which would be more similar to H.264/MPEG-4 AVC.

In the HEVC specification, there are several syntax elements that allow various tools to be configured or enabled. Among these are parameters that specify the minimum and maximum coding block size, transform block size, and transform hierarchy depth. There are also flags to turn tools such as temporal motion vector prediction (TMVP), AMP, SAO and transform skip (TS) on or off. By setting these parameters, the contribution of these tools to the coding performance improvements of HEVC can be gauged.

For the following experiments, the test sequences from classes A to E specified in the appendix and the coding conditions as defined in [32] were used. HEVC test model 8 software HM-8.0 [24] was used for these specific experiments. Two coding structures were investigated – one suitable for entertainment applications with random access support and one for interactive applications with low-delay constraints.

The following tables show the effects of constraining or turning off tools defined in the HEVC Main Profile. In doing so, there will be an increase in bit rate, which is an indication of the benefit that the tool brings. The reported percentage difference in the encoding and decoding time is an indication of the amount of processing that is needed by the tool. Note that this is not suggested to be a reliable measure of the complexity of the tool in an optimized hardware or software based encoder or decoder – but may provide some rough indication.

TABLE I
DIFFERENCE IN BIT RATE FOR EQUAL PSNR RELATIVE TO HEVC MP WHEN SMALLER MAXIMUM CODING BLOCK SIZES WERE USED INSTEAD OF 64×64 CODING BLOCKS.

	Entertainment applications		Interactive applications	
	Maximum coding unit size		Maximum coding unit size	
	32×32	16×16	32×32	16×16
Class A	5.7%	28.2%	–	–
Class B	3.7%	18.4%	4.0%	19.2%
Class C	1.8%	8.5%	2.5%	10.3%
Class D	0.8%	4.2%	1.3%	5.7%
Class E	–	–	7.9%	39.2%
Overall	2.2%	11.0%	3.7%	17.4%
Enc. Time	82%	58%	83%	58%
Dec. Time	111%	160%	113%	161%

Table I compares the effects of setting the maximum coding block size for luma to 16×16 or 32×32 samples, versus the 64×64 maximum size allowed in the HEVC Main Profile. These results show that though the encoder spends less time searching and deciding on the CB sizes, there is a significant penalty in coding efficiency when the maximum block size is limited to 32×32 or 16×16 samples. It can also be seen that the benefit of larger block sizes is more significant for the higher-resolution sequences as well as for sequences with sparse content such as the class E sequences. An interesting effect on the decoder side is that when larger block sizes are used, the decoding time is reduced, as smaller block sizes require more decoding time in the HM implementation.

Table II compares the effects of setting the maximum transform block size to 8×8 and 16×16, versus the 32×32 maximum size allowed in HEVC MP. The results show the same trend as constraining the maximum coding block sizes. However, the percentage bit rate penalty is smaller, since constraining the maximum coding block size also indirectly constrains the maximum transform size while the converse is not true. The amount of the reduced penalty shows that there are some benefits from using larger coding units that are not simply due to the larger transforms. It is however noted that constraining the transform size has a more significant effect on the chroma components than the luma component.

TABLE II
DIFFERENCE IN BIT RATE FOR EQUAL PSNR RELATIVE TO HEVC MP WHEN SMALLER MAXIMUM TRANSFORM BLOCK SIZES ARE USED INSTEAD OF 32×32 TRANSFORM BLOCKS.

	Entertainment applications		Interactive applications	
	Maximum transform size		Maximum transform size	
	16×16	8×8	16×16	8×8
Class A	3.9%	12.2%	–	–
Class B	2.4%	9.3%	2.7%	9.7%
Class C	1.0%	4.2%	1.5%	5.5%
Class D	0.4%	2.4%	0.5%	3.1%
Class E	–	–	3.8%	10.6%
Overall	1.3%	5.4%	2.1%	7.2%
Enc. Time	94%	87%	96%	90%
Dec. Time	99%	101%	99%	101%

HEVC allows the transform block size in a coding unit to be selected independently of the prediction block size (with few exceptions). This is controlled through the residual quadtree (RQT), which has a selectable depth. Table III compares the effects of setting the maximum transform hierarchy depth to 1 and 2 instead of 3, the value used in the common test conditions [32]. It shows that some savings in the encoding decision time can be made for a modest penalty in coding efficiency for all classes of test sequences. However, there is no significant impact on the decoding time.

TABLE III
DIFFERENCE IN BIT RATE FOR EQUAL PSNR RELATIVE TO HEVC MP WHEN SMALLER MAXIMUM RQT DEPTHS WERE USED INSTEAD OF A DEPTH OF 3.

	Entertainment applications		Interactive applications	
	Max RQT depth		Max RQT depth	
	2	1	2	1
Class A	0.3%	0.8%	–	–
Class B	0.4%	1.1%	0.5%	1.4%
Class C	0.4%	1.1%	0.5%	1.5%
Class D	0.3%	1.1%	0.4%	1.4%
Class E	–	–	0.3%	0.8%
Overall	0.3%	1.0%	0.4%	1.3%
Enc. Time	89%	81%	91%	85%
Dec. Time	99%	98%	101%	100%

Table IV shows the effects of turning off TMVP, SAO, AMP, and TS in the HEVC MP. The resulting bit rate increase is measured by averaging over all classes of sequences tested. Bit rate increases of 2.5% and 1.6% were measured when disabling TMVP and SAO, respectively, for the entertainment application scenario. For the interactive application scenario, the disabling of TMVP or SAO tool yielded a bit rate increase of 2.5%. It should be noted that SAO has a larger impact on the subjective quality than on the PSNR. Neither of these tools has a significant impact on encoding or decoding time. When the AMP tool is disabled, bit rate increases of 0.9% and 1.2% were measured for the entertainment and interactive applications scenario, respectively. The significant increase in encoding time can be attributed to the additional motion search and decision that is needed for AMP. Disabling the TS tool does not change the coding efficiency. It should, however, be noted that the TS tool is most effective for content such as computer screen capture and overlays. For such content, disabling of the TS tool shows bit rate increases of 7.3% and 6.3% for the entertainment and interactive application scenarios, respectively.

TABLE IV
DIFFERENCE IN BIT RATE FOR EQUAL PSNR RELATIVE TO HEVC MP WHEN THE TMVP, SAO, AMP, AND TS TOOLS ARE TURNED OFF.

	Entertainment applications				Interactive applications			
	tools disabled in MP				tools disabled in MP			
	TMVP	SAO	AMP	TS	TMVP	SAO	AMP	TS
Class A	2.6%	2.4%	0.6%	0.0%	–	–	–	–
Class B	2.2%	2.4%	0.7%	0.0%	2.5%	2.6%	1.0%	0.0%
Class C	2.4%	1.7%	1.1%	0.1%	2.8%	2.9%	1.1%	0.1%
Class D	2.7%	0.5%	0.9%	0.1%	2.4%	1.3%	1.2%	0.0%
Class E	–	–	–	–	2.4%	3.3%	1.7%	−0.1%
Overall	2.5%	1.6%	0.9%	0.0%	2.5%	2.5%	1.2%	0.0%
Enc. Time	99%	100%	87%	95%	101%	101%	88%	96%
Dec. Time	96%	97%	99%	98%	96%	98%	100%	99%

Results for other tools of HEVC that yield improvements relative to H.264/MPEG-4 AVC (including merge mode, intra prediction, and motion interpolation filter) are not provided here, and the reader is referred to [33].

C. *Results in comparison to previous standards*

For comparing the coding efficiency of HEVC with that of prior video coding standards, we performed coding experiments for the two different scenarios of entertainment and interactive applications. The encoding strategy described in sec. III has been used for all investigated standards. For HEVC, the described encoder control is the same as the one implemented in the HM-8.0 reference software [24], so this software has been used unmodified. For the other standards, we integrated the described encoder control into older encoder implementations. The following codecs have been used as basis: The MPEG Software Simulation Group Software version 1.2 [34] for H.262/MPEG-2 Video, the H.263 codec of the University of British Columbia Signal Processing and Multimedia Group (see [13]), a Fraunhofer HHI implementation of MPEG-4 Visual, and the JSVM software[1] version 9.18.1 [35] for H.264/MPEG-4 AVC. All encoders use the same strategies for mode decision, motion estimation, and quantization. These encoders show significantly improved coding efficiency relative to publicly available reference implementations or the encoder versions that were used in [13].

For HEVC, all coding tools specified in the draft HEVC Main Profile are enabled. For the other tested video coding standards, we selected the profiles and coding tools that provide the best coding efficiency for the investigated scenarios.

[1] The JM 18.4 encoder [36] or the modified JM 18.2, which was used for the comparison in sec. V, provide very similar coding efficiency as our modified JSVM version, but differ in some details from the HM encoder control.

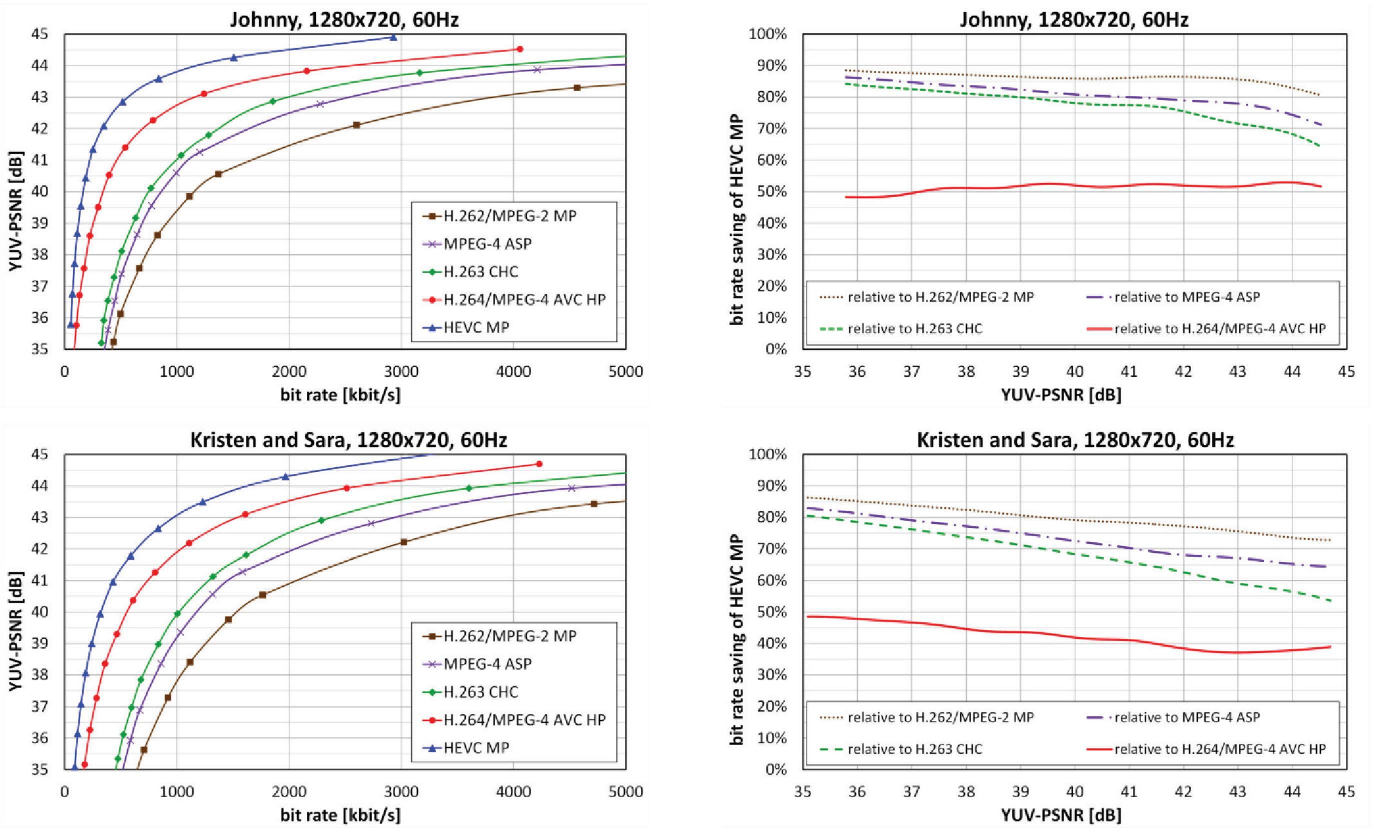


Fig. 1. Selected rate–distortion curves and bit rate saving plots for interactive applications.

The chosen profiles are the H.262/MPEG-2 Main Profile (MP), the H.263 Conversational High Compression (CHC) profile for the interactive scenario and the H.263 High Latency profile (HLP) for the entertainment scenario, the MPEG-4 Advanced Simple Profile (ASP), and the H.264/MPEG-4 AVC High Profile (HP).

Each test sequence was coded at twelve different bit rates. For H.264/MPEG-4 AVC and HEVC, the quantization parameter QP_I for intra pictures was varied in the range from 20 to 42, inclusive. For H.262/MPEG-2 MP and MPEG-4 ASP, the quantization parameters for intra pictures were chosen in a way that the resulting quantization step sizes are approximately the same as for H.264/MPEG-4 AVC and HEVC. The quantization parameters for non-intra pictures are set relative to QP_I using a deterministic approach that is basically the same for all tested video coding standards. In order to calculate bit rate savings for one codec relative to another, the rate–distortion curves were interpolated in the logarithmic bit rate domain using cubic splines with the “not-a-knot” condition at the border points. Average bit rate savings are calculated by numerical integration with 1000 equally sized subintervals.

1) *Interactive applications*

The first experiment addresses interactive video applications, such as video conferencing. We selected six test sequences with typical video conferencing content, which are the sequences of classes E and E' listed in the appendix.

Since interactive applications require a low coding delay, all pictures were coded in display order, where only the first picture is coded as an intra picture and all subsequent pictures are temporally predicted only from reference pictures in the past in display order. For H.262/MPEG-2 Video and MPEG-4 Visual, we employed the IPPP coding structure, where the quantization step size for P pictures was increased by about 12% relative to that for I pictures. The syntax of H.263, H.264/MPEG-4 AVC, and HEVC supports low-delay coding structures that usually provide an improved coding efficiency. Here we used dyadic low-delay hierarchical prediction structures with groups of 4 pictures (cp. [17]). While for H.263 and H.264/MPEG-4 AVC all pictures are coded with P slices, for HEVC, all pictures are coded with B slices. For H.264/MPEG-4 AVC and HEVC, which both support low-delay coding with P or B slices, we selected the slice coding type that provided the best coding efficiency (P slices for H.264/MPEG-4 AVC and B slices for HEVC). The quantization step size for the P or B pictures of the lowest hierarchy level is increased by about 12% relative to that for I picture, and it is further increased by about 12% from one hierarchy level to the next. For H.263, H.264/MPEG-4 AVC, and HEVC, the same four previously coded pictures are used as active reference pictures. Except for H.262/MPEG-2 Video, which does not support slices that cover more than one macroblock row, all pictures are coded as a single slice. For H.262/MPEG-2 Video, one slice per macroblock row is used. Inverse transform mismatches for H.262/MPEG-2 Video, H.263, and MPEG-4 Visual are avoided, since the used decoders implement exactly the same transform as the corresponding encoder. In practice, where this cannot be guaran-

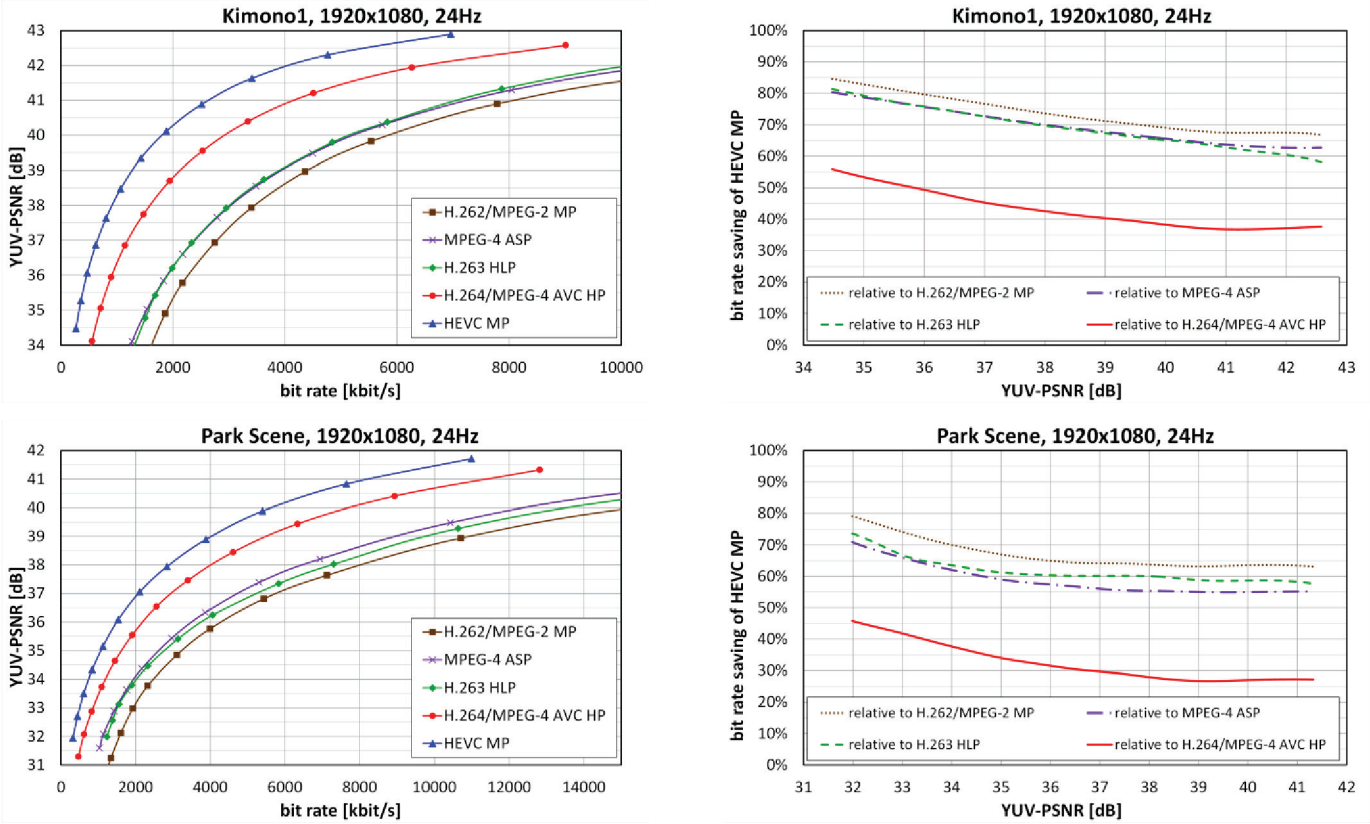


Fig. 2. Selected rate–distortion curves and bit rate saving plots for entertainment applications.

teed, the PSNR values and subjective quality for these standards would be reduced; and intra macroblocks would need to be inserted periodically in order to limit the mismatch accumulation.

TABLE V
AVERAGE BIT RATE SAVINGS FOR EQUAL PSNR
FOR INTERACTIVE APPLICATIONS.

Encoding	Bit rate savings relative to:			
	H.264/MPEG-4 AVC HP	H.263 CHC	MPEG-4 ASP	MPEG-2/ H.262 MP
HEVC MP	40.3%	67.9%	72.3%	80.1%
H.264/MPEG-4 AVC HP	–	46.8%	54.1%	67.0%
H.263 CHC	–	–	13.2%	37.4%
MPEG-4 ASP	–	–	–	27.8%

In Fig. 1, rate–distortion curves are depicted for two selected sequences, in which the PSNR_{YUV} as defined in sec. IV.A is plotted as a function of the average bit rate. This figure additionally shows plots that illustrate the bit rate savings of HEVC relative to H.262/MPEG-2 MP, H.263 CHC, MPEG-4 ASP, and H.264/MPEG-4 AVC HP as a function of the PSNR_{YUV}. In the diagrams, the PSNR_{YUV} is denoted as YUV-PSNR. The average bit rate savings between the different codecs, which are computed over the entire test set and the investigated quality range, are summarized in Table V. These results indicate that the emerging HEVC standard clearly outperforms its predecessors in terms of coding efficiency for interactive applications. The rate savings for the low bit rate range are generally somewhat higher than the average savings given in Table V, which becomes evident from the plots in the right column of Fig. 1.

2) *Entertainment applications*

Besides interactive applications, one of the most promising application areas for HEVC is the coding of high-resolution video with entertainment quality. For analyzing the potential of HEVC in this application area, we have selected a set of five full HD and four WVGA test sequences, which are listed as class B and C sequences in the appendix.

In contrast to our first experiment, the delay constraints are relaxed for this application scenario. For H.264/MPEG-4 AVC and HEVC, we used dyadic high-delay hierarchical prediction structures (cf. [17]) with groups of 8 pictures, where all pictures are coded as B pictures except at random access refresh points (where I pictures are used). This prediction structure is characterized by a structural delay of 8 pictures and has been shown to provide an improved coding efficiency compared to IBBP coding. Similarly as for the first experiment, the quantization step size is increased by about 12% (QP increase by 1) from one hierarchy level to the next, and the quantization step size for the B pictures of the lowest hierarchy level is increased by 12% relative to that of the intra pictures. The same four active reference pictures are used for H.264/MPEG-4 AVC and HEVC. H.262/MPEG-2 Video, H.263, and MPEG-4 Visual do not support hierarchical prediction structures. Here we used a coding structure where three B pictures are inserted between each two successive P pictures. The usage of three B pictures ensures that the intra pictures are inserted at the same locations as for the H.264/MPEG-4 AVC and HEVC configurations, and it slightly improves the coding efficiency in comparison to the typical coding structure with two B pictures. The quantization step sizes were increased by about 12% from I to P pictures and from P to B pictures. For H.263, four active reference pictures are used for both the P and B pictures.

For all tested codecs, intra pictures are inserted in regular time intervals of about one second, at exactly the same time instances. Such frequent periodic intra refreshes are typical in entertainment-quality applications in order to enable fast random access – e.g., for channel switching. In order to enable clean random access, pictures that follow an intra picture in both coding and display order are not allowed to reference any picture that precedes the intra picture in either coding or display order. However, pictures that follow the intra picture in coding order but precede it in display order are generally allowed to use pictures that precede the intra picture in coding order as reference pictures for motion-compensated prediction. This structure is sometimes referred to as “open GOP”, where a GOP is a “group of pictures” that begins with an I picture.

TABLE VI
AVERAGE BIT RATE SAVINGS FOR EQUAL PSNR
FOR ENTERTAINMENT APPLICATIONS.

Encoding	Bit rate savings relative to:			
	H.264/MPEG-4 AVC HP	MPEG-4 ASP	H.263 HLP	MPEG-2/ H.262 MP
HEVC MP	35.4%	63.7%	65.1%	70.8%
H.264/MPEG-4 AVC HP	–	44.5%	46.6%	55.4%
MPEG-4 ASP	–	–	3.9%	19.7%
H.263 HLP	–	–	–	16.2%

The diagrams in Fig. 2 show rate–distortion curves and bit rate saving plots for two typical examples of the tested sequences. The bit rate savings results, averaged over the entire set of test sequences and the examined quality range, are summarized in Table VI. As for the previous case, HEVC provides significant gains in term of coding efficiency relative to the older video coding standards. As can be seen in the plots in Fig. 2, the coding efficiency gains for the lower bit rate range are again generally higher than the average results reported in Table VI.

V. PRELIMINARY INVESTIGATION OF THE HEVC REFERENCE IMPLEMENTATION COMPARED TO H.264/MPEG-4 AVC USING SUBJECTIVE QUALITY

A. Laboratory and test setup

The laboratory for the subjective assessment was set up following ITU-R Rec. BT.500 [37], except for the section on the displays and video server. A 50-inch Panasonic professional plasma display (TH-50PF11KR) was used in its native resolution of 1920×1080 pixels. The video display board was a Panasonic Dual Link HD-SDI input module (TY-FB11DHD). The uncompressed video recorder/player was a UDR-5S by Keisoku Giken Co., Ltd., controlled using a DellPrecisionT3500.

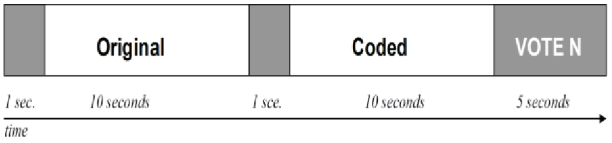


Fig. 3. DSIS basic test cell.

DSIS (Double Stimulus Impairment Scale) as defined in the HEVC Call for Proposals [38] was used for the evaluation of the quality (rather than of the impairment). Hence, a quality rating scale made of 11 levels was adopted, ranging from “0” (lowest quality) to “10” (highest quality).

The structure of the Basic Test Cell (BTC) of the DSIS method consists of two consecutive presentations of the sequence under test. First the original version of the video sequence is displayed, followed immediately by the decoded sequence. Then a message is shown for 5 seconds asking the viewers to vote (see Fig. 3). The presentation of the video clips is preceded by a mid-level gray screen for a duration of one second.

Each test session comprised tests on a single test sequence and lasted approximately 8 minutes. A total of 9 test sequences, listed as class B and C in the appendix, were used in the subjective assessment. The total number of test subjects was 24. The test subjects were divided into groups of four in each test session, seated in a row. A viewing distance of $2H$ was used in all tests, where H is the height of the video on the plasma display.

B. Codecs tested and coding conditions

In the subjective assessment, the test sequences for H.264/MPEG-4 AVC HP were encoded using the JM 18.2 codec with the encoder modifications as described in [39][40]. The test sequences for the HEVC MP were encoded using the HM-5.0 software [41]. It should be noted that the HEVC MP configuration by the time of HM-5.0 was slightly worse in performance than HM-8.0 [24] and also did not include AMP.

The same random access coding structure was used in all test sequences. Quantization parameter (QP) values of 31, 34, 37 and 40 were selected for the HEVC MP. For H.264/MPEG-4 AVC HP, QP values of 27, 30, 33 and 36 were chosen. It was confirmed in a visual pre-screening that these settings resulted in decoded sequences of roughly comparable subjective quality and the bit rate reductions for the HEVC MP encodings ranged from 48% to 65% (53% on average) relative to the corresponding H.264/MPEG-4 AVC HP bit rates.

C. Results

Fig. 4 shows the result of the formal subjective assessment. The mean opinion score (MOS) values were computed from the votes provided by the subjects for each test point. The 95% confidence interval was also calculated and represented as vertical error bars on the graphs. As can be seen from the example, corresponding points have largely overlapping confidence intervals, indicating that the quality of the sequences would be measured within these intervals again with 95% probability. This confirms that the test sequences encoded

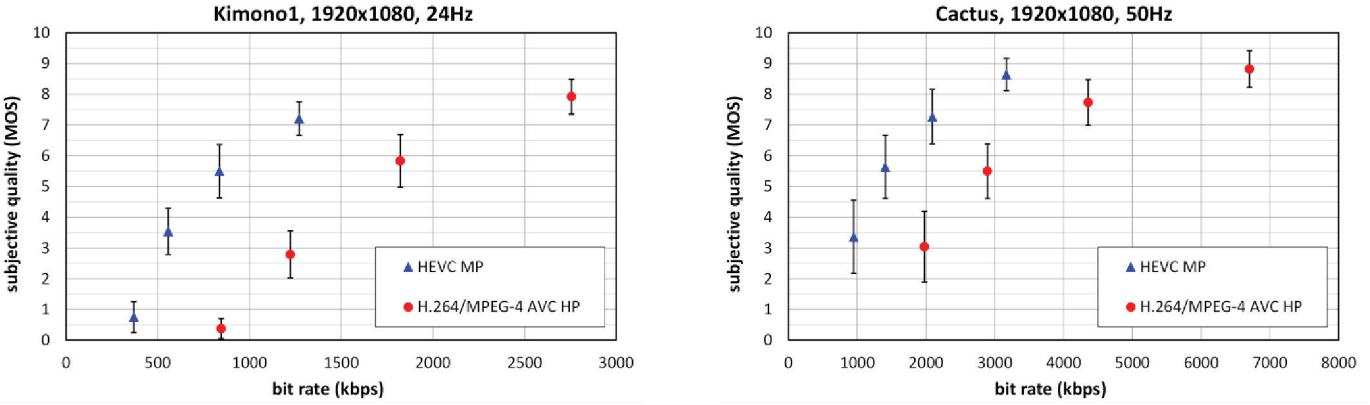


Fig. 4. Mean opinion score (MOS) for test sequences plotted against bit rate.

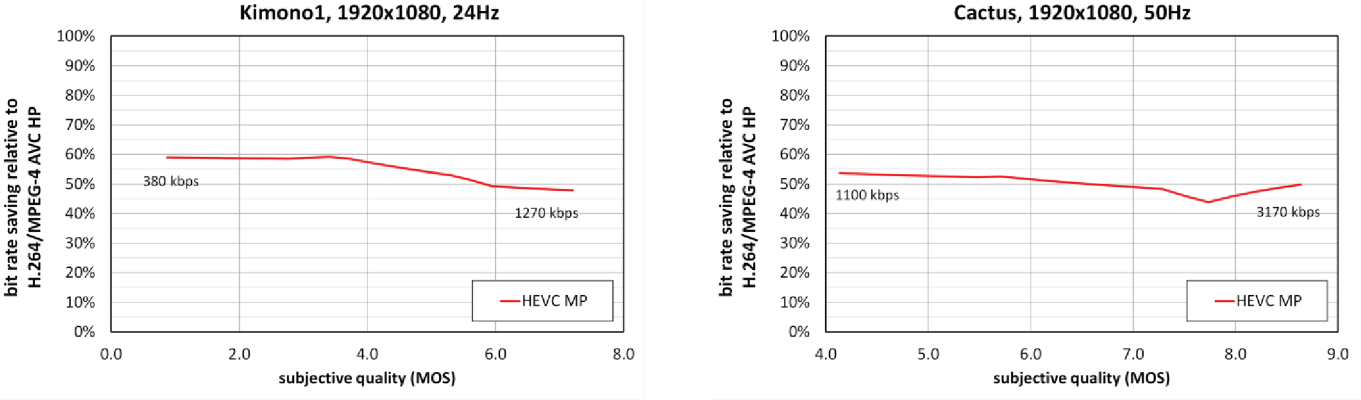


Fig. 5. Bit rate savings as a function of subjective quality.

with HEVC at an average of 53% lower bit rate than the H.264/MPEG-4 AVC HP encodings achieved approximately the same subjective quality.

TABLE VII
AVERAGE BIT RATE SAVINGS FOR ENTERTAINMENT APPLICATION SCENARIO BASED ON SUBJECTIVE ASSESSMENT RESULTS.

Sequences	Bit rate savings of HEVC MP relative to H.264/MPEG-4 AVC HP
BQ Terrace	63.1%
Basketball Drive	66.6%
Kimono1	55.2%
Park Scene	49.7%
Cactus	50.2%
BQ Mall	41.6%
Basketball Drill	44.9%
Party Scene	29.8%
Race Horses	42.7%
Average	49.3%

D. *Further processing of the results*

The subjective test results were further analyzed to obtain a finer and more precise measure of the coding performance gains of the HEVC standard. There are a set of four MOS values per sequence per codec. By linearly interpolating between these points, the intermediate MOS values and the corresponding bit rates for each of the codecs can be approximated. By comparing these bit rates at the same MOS values, the bit rate savings achieved by HEVC relative to H.264/MPEG-4 AVC can be calculated for any given MOS values. An example is shown in Fig. 5. These graphs show the bit rate savings for the HEVC MP relative to the H.264/MPEG-4 AVC HP at different MOS values. The corresponding bit rates for the HEVC MP are also shown at the two ends of the curve.

By integrating over the whole range of overlapping MOS values, the average bit rate savings per sequence can be obtained. Table VII shows the computed bit rate savings of the HEVC MP relative to H.264/MPEG-4 AVC HP. The savings ranges from around 30% to nearly 67%, depending on the video sequence. The average bit rate reduction over all the sequences tested was 49.3%.

VI. SUMMARY AND CONCLUSIONS

The results documented in this paper indicate that the emerging HEVC standard can provide a significant amount of increased coding efficiency compared to previous standards, including H.264/MPEG-4 AVC. The syntax and coding structures of the various tested standards were explained, and the associated Lagrangian-based encoder optimization has been described. Special emphasis has been given to the various settings and tools of HEVC that are relevant to its coding efficiency. Measurements were then provided for their assessment. PSNR vs. bit rate measurements have been presented comparing the coding efficiency of the capabilities of HEVC, H.264/MPEG-4 AVC, MPEG-4 Visual, H.263, and H.262/MPEG-2 Video when encoding using the same Lagrangian-based optimization techniques. Finally, results of subjective tests were provided comparing HEVC and H.264/MPEG-4 AVC, and indicating that a bit rate reduction can be achieved for the example video test set by about 50%. The subjective benefit for HEVC seems to exceed the benefit measured using PSNR, and the benefit is greater for low bit rates, higher-resolution video content and low-delay application encodings. These results generally agree with the preliminary coding efficiency evaluations of HEVC that have reported in other studies such as [39], [40], and [42]–[46], although the subjective estimate here may be generally slightly more conservative than in prior studies, due to our use of stronger encoding optimization techniques in the encodings for the prior standards.

Software and data for reproducing selected results of this study can be found at ftp://ftp.hhi.de/ieee-tcsvt/2012/.

ACKNOWLEDGMENT

The authors would like to thank Vittorio Baroncini (Fondazione Ugo Bordoni) for conducting and coordinating various subjective tests throughout the development of the standard, including the recent tests reported in [42] and the evaluation of the responses to the HEVC call for proposals [47], Anthony Joch (Avvasi Inc.) and Faouzi Kossentini (eBrisk Video) for providing the source code of an H.263++ implementation, Tobias Hinz (Fraunhofer HHI) for providing the source code of an MPEG-4 Visual implementation, Bin Li (University of Science and Technology of China) and Jizheng Xu (Microsoft Corp.) for providing the JM18.2 software modifications [39], and Junya Takiue, Akira Fujibayashi and Yoshinori Suzuki (NTT DOCOMO, Inc) for conducting the preliminary HEVC subjective tests presented in section V.

TABLE VIII
TEST SEQUENCES USED IN THE COMPARISONS.

class	resolution in luma samples	length	sequence	frame rate
A	2560×1600	5 s	Traffic	30 Hz
			People On Street	30 Hz
			Nebuta	60 Hz
			Steam Locomotive	60 Hz
B	1920×1080	10 s	Kimono	24 Hz
			Park Scene	24 Hz
			Cactus	50 Hz
			BQ Terrace	60 Hz
			Basketball Drive	50 Hz
C	832×480	10 s	Race Horses	30 Hz
			BQ Mall	60 Hz
			Party Scene	50 Hz
			Basketball Drill	50 Hz
D	416×240	10 s	Race Horses	30 Hz
			BQ Square	60 Hz
			Blowing Bubbles	50 Hz
			Basketball Pass	50 Hz
E	1280×720	10 s	Four People	60 Hz
			Johnny	60 Hz
			Kristen And Sara	60 Hz
E'	1280×720	10 s	Vidyo 1	60 Hz
			Vidyo 2	60 Hz
			Vidyo 3	60 Hz

APPENDIX
TEST SEQUENCES

Details about the test sequences and sequences classes that are used for the comparisons in the paper are summarized in Table VIII. The sequences were captured with state-of-the-art cameras. All sequences are progressively scanned and use the YUV (YC_BC_R) 4:2:0 color format with 8 bits per color sample.

REFERENCES

[1] G. J. Sullivan and J.-R. Ohm, "Recent Developments in Standardization of High Efficiency Video Coding (HEVC)," *SPIE Applications of Digital Image Processing XXXIII,* San Diego, USA, *Proc. SPIE*, vol. 7798, paper 7798-30, Aug. 2010.

[2] T. Wiegand, J.-R. Ohm, G. J. Sullivan, W.-J. Han, R. Joshi, T. K. Tan, and K. Ugur, "Special Section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) Standardization," *Special Section of IEEE Trans. Circuits and Systems for Video Tech. on the Joint Call for Proposals on High Efficiency Video Coding (HEVC)*, vol. 20, no. 12, pp. 1661–1666, Dec. 2010.

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits and Systems for Video Tech.*, this issue.

[4] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 8," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-J1003, Stockholm, Sweden, July, 2012.

[5] ITU-T and ISO/IEC JTC 1, "Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video," ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2), version 1: 1994.

[6] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard*, Kluwer Academic Publishers, 2000.

[7] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*, Kluwer Academic Publishers, 2002.

[8] ITU-T, *Video Coding for Low Bitrate Communication*, ITU-T Rec. H.263, version 1, 1995, version 2, 1998, version 3, 2000.

[9] ISO/IEC JTC 1, *Coding of Audio-Visual Objects – Part 2: Visual*, ISO/IEC 14496-2 (MPEG-4 Visual), version 1: 1999, version 2: 2000, version 3: 2004.

[10] ITU-T and ISO/IEC JTC 1, *Advanced Video Coding for generic audio-visual services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), version 1: 2003, version 2: 2004, versions 3, 4: 2005, versions 5, 6: 2006, versions 7, 8: 2007, versions 9, 10, 11: 2009, versions 12, 13: 2010, versions 14, 15: 2011, version 16: 2012.

[11] G. J. Sullivan and T. Wiegand, "Video Compression – From Concepts to the H.264/AVC Standard," *Proceedings of the IEEE*, Vol. 93, no. 1, pp. 18–31, Jan. 2005.

[12] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Systems for Video Tech.*, Vol. 13, No. 7, pp. 560–576, July 2003.

[13] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," *IEEE Trans. Circuits and Systems for Video Tech.*, Vol. 13, No. 7, pp. 688–703, July 2003.

[14] ITU-T, *Video Codec for Audiovisual Services at p×64 Kbit/s*, ITU-T Rec. H.261, version 1: 1990, version 2: 1993.

[15] I. E. Richardson, *H.264 und MPEG-4 Video Compression*, John Wiley & Sons, 2003.

[16] ISO/IEC JTC 1, *Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s – Part 2: Video*, ISO/IEC 11172-2 (MPEG-1), 1993.

[17] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 17, pp. 1103–1120, Sep. 2007.

[18] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, pp. 399–417, May-June 1963.

[19] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445–1453, Sep. 1988.

[20] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 74–90, Nov. 1998.

[21] S.-W. Wu and A. Gersho, "Enhanced video compression with standardized bit stream syntax," *IEEE Intl. Conf. on Acoust., Systems, and Signal* Proc., Vol. I, pp. 103–106, Apr. 1993.

[22] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 6, pp. 182–190, Apr. 1996.

[23] G. J. Sullivan and R. L. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," in *Proc. of GLOBECOM'91*, pp. 85–90, Dec. 1991.

[24] Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, "HM-8.0 reference software," https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-8.0/.

[25] S. Saponara, C. Blanch, K. Denolf, and J. Bormans, "The JVT advanced video coding standard: Complexity and performance analysis on a tool-by-tool basis," Packet Video Workshop, Nantes, France, Apr. 2003.

[26] M. Flierl and B. Girod, "Generalized B pictures and the draft H.264/AVC video compression standard," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 13, pp. 587–597, July 2003.

[27] T. Wiegand and B. Andrews, "An improved H.263 coder using rate-distortion optimization," VCEG, document Q15-D-13, Mar. 1998.

[28] K. Ramchandran and M. Vetterli, "Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility," *IEEE Trans. Image Processing*, vol. 3, no. 5, pp. 700–704, Sep. 1994.

[29] E.-H. Yang, X. Yu, "Rate distortion optimization for H.264 interframe coding: A general framework and algorithms," *IEEE Trans. Image Processing*, vol. 16, pp. 1774–1784, July 2007.

[30] M. Karczewicz, Y. Ye, and I. Chong, "Rate distortion optimized quantization," ITU-T SG 16/Q 6, document VCEG-AH21, Jan. 2008.

[31] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD curves", ITU-T SG 16/Q 6, document VCEG-M33, Austin, Texas, USA, Apr. 2001.

[32] F. Bossen., "Common conditions and software reference configurations", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-H1100, San José, USA, Feb. 2012.

[33] T.K. Tan, A. Fujibayashi, Y. Suzuki and J. Takiue, "[AHG 8] Objective and subjective evaluation of HM5.0," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-H0116, San José, USA, Feb. 2012.

[34] MPEG Software Simulation Group Software, version 1.2, available at http://www.mpeg.org/MPEG/video/mssg-free-mpeg-software.html.

[35] Joint Scalable Video Model Software, version 9.19.15, available as described in the document JVT-AF013, Joint Video Team, Nov. 2009.

[36] H.264/MPEG-4 AVC Reference Software, Joint Model 18.4, available at http://iphome.hhi.de/suehring/tml/download/jm18.4.zip.

[37] ITU-R, *Methodology for the subjective assessment of the quality of television pictures,* ITU-R Rec. BT.500-11, Feb. 2006.

[38] ITU-T VCEG and ISO/IEC MPEG, "Joint Call for Proposals on Video Compression Technology," document VCEG-AM91 and WG11 N11113, Kyoto, Japan, Jan. 2010.

[39] B. Li, G. J. Sullivan, and J. Xu, "Comparison of Compression Performance of HEVC Working Draft 7 with AVC High Profile", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-J0236, Stockholm, Sweden, July 2012.

[40] B. Li, G. J. Sullivan, and J. Xu, "Compression Performance of High Efficiency Video Coding (HEVC) Working Draft 4," *IEEE Intl. Conf. on Circuits and Systems* (ISCAS), pp. 886–889, May 2012.

[41] Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, "HM-5.0 reference software," https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-5.0/.

[42] Y. Zhao, "Coding efficiency comparison between HM5.0 and JM16.2 based on PQI, PSNR and SSIM," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-H0063, San José, USA, Feb. 2012.

[43] T. K. Tan, A. Fujibayashi, Y. Suzuki, J. Takiue, "Objective and subjective evaluation of HM5.0," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-H0116, San José, USA, Feb. 2012.

[44] V. Baroncini, G. J. Sullivan, and J.-R. Ohm, "Report on Preliminary Subjective Testing of HEVC Compression Capability," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-H1004, San José, USA, Feb. 2012.

[45] P. Hanhart, M. Rerabek, F. De Simone, T. Ebrahimi, "Subjective Quality Evaluation of the Upcoming HEVC Video Compression Standard," *SPIE Applications of Digital Image Processing XXXV*, San Diego, USA, Proc. SPIE, Vol. 8499, paper 8499-30, Aug. 2012.

[46] M. Horowitz, F. Kossentini, N. Mahdi, S. Xu, H. Guermazi, H. Tmar, B. Li, G. J. Sullivan, and J. Xu, "Informal Subjective Quality Comparison of Video Compression Performance of the HEVC and H.264 / MPEG-4 AVC Standards for Low-Delay Applications," *SPIE Applications of Digital Image Processing XXXV*, San Diego, USA, Proc. SPIE, Vol. 8499, paper 8499-31, Aug. 2012.

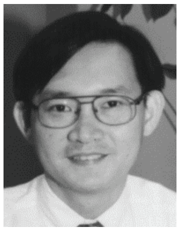
[47] V. Baroncini, G. J. Sullivan, and J.-R. Ohm, "Report of subjective testing of responses to Joint Call for Proposals (CfP) on video coding technology for High Efficiency Video Coding (HEVC)," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JCTVC-A204, Dresden, Germany, Apr. 2010.

Jens-Rainer Ohm (M'92). See biography on page [INSERT PAGE NUMBER] of this issue.

Gary J. Sullivan (S'83–M'91–SM'01–F'06). See biography on page [INSERT PAGE NUMBER] of this issue.

Heiko Schwarz received the Dipl.-Ing. degree in electrical engineering and the Dr.-Ing. degree, both from the University of Rostock, Rostock, Germany, in 1996 and 2000, respectively.

In 1999, he joined the Image and Video Coding Group, Fraunhofer Institute for Telecommunications–Heinrich Hertz Institute, Berlin, Germany. Since then, he has contributed successfully to the standardization activities of the ITU-T Video Coding Experts Group (ITU-T SG16/Q.6-VCEG) and the ISO/IEC Moving Pictures Experts Group (ISO/IEC JTC 1/SC 29/WG 11 – MPEG). During the development of the scalable video coding extension of H.264/MPEG-4 AVC, he co-chaired several ad hoc groups of the Joint Video Team of ITU-T VCEG and ISO/IEC MPEG investigating particular aspects of the scalable video coding design. He has been appointed as a Co-Editor of ITU-T Rec. H.264 and ISO/IEC 14496-10 and as a Software Coordinator for the SVC reference software.



Thiow Keng Tan (S'89–M'94–SM'03) received the Bachelor of Science and Bachelor of Electrical and Electronics Engineering degrees from Monash University, Australia in 1987 and 1989, respectively. He later received the Ph.D. degree in Electrical Engineering in 1994 from the same university.

He currently consults for NTT DOCOMO, Inc., Japan. He is an active participant at the video subgroup of the ISO/IEC JCT1/SC29/WG11 Moving Picture Experts Group (MPEG), the ITU-T SG16 Video Coding Experts Group (VCEG) as well as the ITU-T/ISO/IEC Joint Video Team (JVT) and the ITU-T/ISO/IEC Joint Collaborative Team for Video Coding (JCT-VC) standardization activities. He has also served on the editorial board of the IEEE Transaction on Image Processing.

Dr. Tan was awarded the Dougles Lampard Electrical Engineering Medal for his Ph.D. thesis and 1st prize IEEE Region 10 Student Paper Award for his final year undergraduate project. He was also awarded three ISO certificates for outstanding contributions to the development of the MPEG-4 standard. He is the inventor in at least 50 granted US patents. His research interest is in the area of image and video coding, analysis and processing.

Thomas Wiegand (M'05–SM'08–F'11). See biography on page [INSERT PAGE NUMBER] of this issue.