**INTERNATIONAL ORGANIZATION FOR STANDARDIZATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 N15074**

**Feb. 2015, Geneva, CH**

|  |  |
| --- | --- |
| **Source** | **Communication** |
| **Title** | **White Paper on HEVC** |
| **Authors** | **Jens-Rainer Ohm, Gary Sullivan** |

# Introduction

High Efficiency Video Coding (HEVC) is a video compression standard developed jointly by ITU-T VCEG and ISO/IEC MPEG through their Joint Collaborative Team on Video Coding (JCT-VC). The first version of HEVC was finalized in January 2013, and the second version in July 2014 provided additional functionality. The HEVC project was launched to achieve major savings for equivalent visual quality relative to the bit rate needed by the widely used AVC [1] standard. For high-resolution video, where such additional compression is most urgently required, implementations of the current draft standard are already meeting or exceeding the targeted goal. This white paper reviews the architecture and building blocks of HEVC, which were carefully selected with regard to compression capability versus complexity and to enable parallelism for the signal processing operations. Given the benefits that HEVC provides, it is likely to become the new primary reference for video compression.

# Background

The history of video compression standardization has been dominated by efforts to maximize compression capability, where an increasing variety of services, the growing popularity of HD video, and the emerging beyond-HD formats (e.g. 4K×2K resolution) are creating even stronger needs for compression capabilities. Mobile devices and tablet PCs will now need to receive and display HD video as well. The advent of stereo and multi-view displays (and corresponding camera systems) further increases the amount of data. As increasing video traffic is seen as a major cause of the “spectrum crunch” that could occur in wireless networks, improved video compression can be a factor to mitigate this as well. Moreover, video now comprises a majority of Internet data traffic worldwide, and its percentage of that traffic is continually increasing.

HEVC has been designed to address essentially all existing applications of previous standards and to particularly focus on two key issues: increased video resolution and increased use of parallel processing architectures. The syntax of HEVC is generic, and its design elements could also be attractive for other application domains that have not used the preceding standards.

# Technology and key features

The video coding layer of HEVC employs essentially the same “hybrid” approach (inter- / intra-picture prediction and 2D transform coding) used in all video compression standards since H.261. A prediction residual is computed from previously decoded information (either previously decoded pictures for inter-picture motion compensated prediction or previously decoded samples from the same picture for intra-picture spatial prediction). The residual is then processed by a block transform, and the transform frequency coefficients are quantized and entropy coded. Side information data such as motion vectors and mode switching parameters are also encoded and transmitted.

Although much of this is similar to the design in prior standards, there are key differences that enable the enhanced compression capability of HEVC, which we discuss below. A more detailed description of the key technical features can be found in [2] and [3], reports about performance and complexity analysis were given in [4] and [5], respectively. The following paragraphs a-h describe features that were defined in version 1 and are also common (in some cases extended) with version 2, whereas features described in i-j were newly defined in version 2.

**a) Coding Tree Units and Coding Tree Block structure:** The core of the coding layer in previous standards was the *macroblock*, consisting of a 16×16 block of luma samples and, in the usual case of “4:2:0” color sampling, two corresponding 8×8 blocks of chroma samples and associated syntax elements. In contrast, the analogous structure in HEVC is the *coding tree unit* (CTU). The CTU consists of *coding tree blocks* (CTBs) for luma and chroma. The size *L*×*L* of a luma CTB can be chosen with *L* = 16, 32, or 64 samples, with the larger sizes typically enabling better compression. CTBs are then partitioned into *coding blocks* (CBs), signaled via a quadtree structure, as illustrated in Fig. 1. Usually, luma and chroma CBs are both split together, where it depends on the color sampling format whether they have the same size or not. A luma CB can be as small as 8×8 (accompanied by two 4×4 chroma CBs in case of 4:2:0 color sampling, for example). One luma CB together with the two corresponding chroma CBs and associated syntax elements is referred to as a *coding unit* (CU).

Below the CU level, there is additional partitioning into *prediction units* (PUs) and a tree of *transform units* (TUs). The decision whether to encode a picture area by *inter-*picture (motion compensated) or *intra-*picture prediction is made at the CU level. The CB shapes are always square. The luma and chroma *prediction blocks* (PBs) within a PU are also always square in the case of intra-picture prediction; however, the shapes used for inter-picture prediction can also be chosen as non-square rectangular blocks.

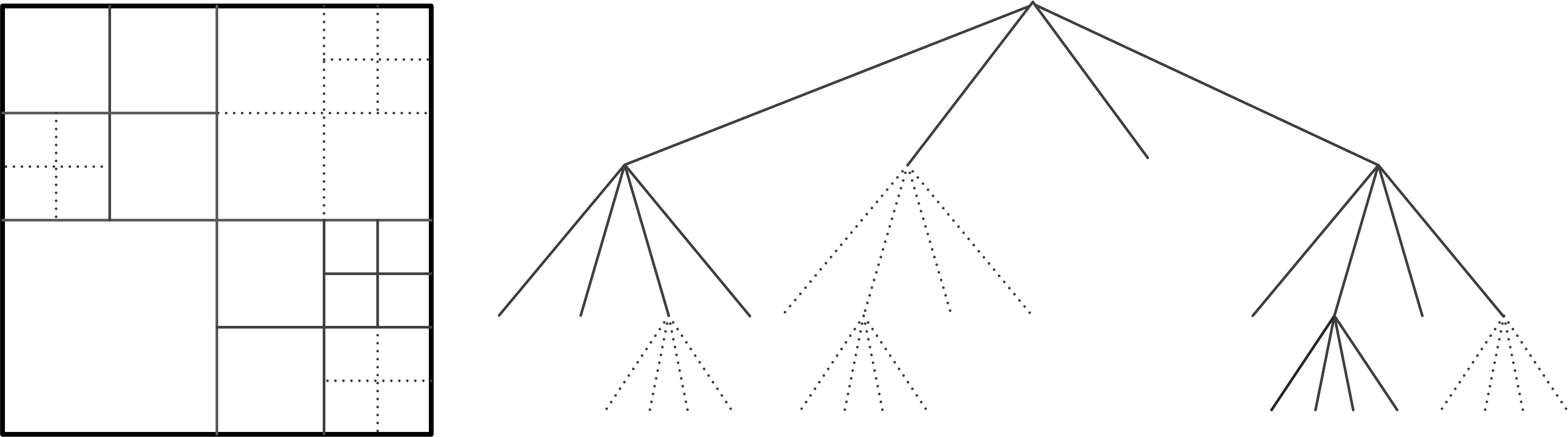


Fig. 1. Subdivision of a 64x64 luma CTB into CBs and TBs. Solid lines indicate CB boundaries and dotted lines indicate TB boundaries. Left: the CTB with its partitioning, right: the corresponding quadtree. In this example, the leaf nodes are each 8×8 in size – although, in general, a TB can actually be as small as 4×4.

**b) Transform Units and Transform Blocks**: The prediction residual is coded using block transforms. A *transform unit* (TU) tree structure has its root at the CU level, where the CBs may be further split into smaller *transform blocks* (TBs). Integer basis functions approximating the *discrete cosine transform* (DCT) are defined for dyadic TB sizes from 4×4 to 32×32 (whereas the largest transform size is 8x8 in AVC). For the 4×4 transform of intra-picture prediction residuals, an integer transform derived from the *discrete sine transform* (DST) is additionally specified, which better matches the statistics of directional intra-picture prediction residual signals. The quantization and coding of transform coefficients is similar to AVC. Coding and decoding of non-zero coefficients is performed by grouping them into 4×4 coefficient groups and scanning the coefficients in each group using a scanning order that is usually diagonal, but changes to horizontal or vertical in the case of small TBs (8x8 and smaller) that are coded using certain directional modes of intra-picture prediction. The position of the last non-zero coefficient in the scanning order is encoded, followed by a “significance map” to identify which coefficients have non-zero values, and then the sign and magnitude of the significant coefficients are signaled.

**c) Motion compensation:** Motion compensation uses quarter-sample precision, and 7-tap or 8-tap filters are used for interpolation of any fractional-sample positions (compared to 6-tap filtering of half-sample positions followed by averaging between two values to form quarter-sample positions in AVC). Similar to AVC, multiple reference pictures are used – with up to 6 stored pictures (or, when using reduced picture resolutions, sometimes up to 16). Per PB, either one or two motion vectors can be transmitted, resulting in uni-predictive or bi-predictive coding, respectively. For the latter case, either ordinary averaging or weighted prediction can be employed. Advanced *motion vector* (MV) coding is used, including derivation of several most probable candidates based on data from adjacent PBs and the co-located region in a reference picture. A new merge mode for MV coding is also defined, signaling the inheritance of MVs from one of the neighboring PBs within the same CTU or from the reference picture. Moreover, compared to AVC, improved “skip” and “direct” motion inference is specified – and in these cases no selection is signaled and the most probable candidate is simply selected and used without modification.

**d) Intra-picture prediction:** Decoded boundary samples from adjacent blocks are used as prediction reference data for spatial prediction in PB regions when inter-picture prediction is not performed. Intra-picture prediction uses 33 directional modes (compared to 8 such modes in AVC), plus DC (flat overall averaging) and planar (surface fitting) prediction modes. Chroma prediction is similar, but uses a simplified selection between fewer modes (horizontal, vertical, planar, DC, the same mode used for luma, or left-downward diagonal). The different intra-picture prediction modes are encoded by deriving most probable modes (e.g. the prediction directions) based on those of previously-decoded neighboring PBs.

**e) Entropy coding:** Five generic binarization schemes are defined for symbol encoding, and it is specified which of these is applied to each type of syntax element. *Context-adaptive binary arithmetic coding* (CABAC) is then used for entropy coding. The basic method is similar to the CABAC scheme in AVC, but has undergone a number of improvements, especially in regard to reducing the number of adaptive coding contexts, increasing the use of fast “bypass” coding, and improving the capability for parallel processing to increase the throughput.

**f) In-loop filtering:** One or two filtering stages can be optionally applied (within the inter-picture prediction loop) before writing the reconstructed picture into the decoded picture buffer. A *deblocking filter* (DBF) is used that is similar to the one in AVC; however the DBF design has been simplified with regard to its decision making and filtering processes and also has been made more friendly to parallel processing. The number of luma edges to be filtered by the DBF in the worst case has been cut in half relative to the filtering in AVC, by limiting the filtering operations of the DBF to operate only on an 8×8 rather than 4×4 block boundary grid. The second stage, called the *sample adaptive offset* (SAO) filter, is a non-linear amplitude mapping. The goal of SAO is to improve the reconstruction of the signal amplitudes by a lookup table mapping that is controlled by the encoder. Two types of SAO operation can be selected for each CTB – the band offset and edge offset modes. Using band offsets, banding artifacts can be reduced by dividing the signal amplitude range into segments and adding offset values that depend on the amplitude category of the value of each reconstructed sample. Edge offsets have a directional “de-ringing” effect, and are controlled by an encoder selection between a vertical, horizontal, downward-diagonal, or upward-diagonal filtering direction. The value of each sample is checked in relation to its directional neighbors to identify regions that have a local minimum or maximum or a local edge – and then an offset value is added that depends on this classification.

**g) Slices, tiles and wavefronts**:A *slice* is a series of CTUs that can be decoded independently from other slices of the same picture (except for in-loop filtering of the edges of the slice, and except for the case of “dependent slices” as described below). A slice can either be an entire picture or a region of a picture. One of the main purposes of slices is re-synchronization after data losses. An example partitioning of a picture into a slice structure is shown in Fig. 2a.

To enable parallel processing and localized access to picture regions, the encoder can partition a picture into rectangular regions called *tiles*. Fig. 2b shows an example. Tiles are also independently decodable but can share some header information when multiple tiles are used within a slice. Alternatively, for finer-granularity packetization of data, an encoder can use multiple slices within a tile (but is not allowed to mix these two behaviors within the same picture). An additional supported form of enabling parallelism is for the encoder to use*wavefront parallel processing* (WPP), in which a slice is divided into rows of CTUs. With WPP, the encoding or decoding of CTUs of each row can begin before processing most of the CTUs of the previous row – thus enabling different processing threads to work on different rows of the picture at the same time, as shown in Fig. 2c. (To minimize the difficulty of making decoders, encoders are prohibited from using WPP when using multiple tiles per picture.) Additionally, to enable packetization at a fine level of granularity, *dependent slices* are also defined, which allows the header information for multiple sequences of CTUs to share the same slice header information, each starting at a tile or wavefront entry point.



Fig. 2. Subdivision of a picture into (a) slices and (b) tiles, and illustration of wavefront parallel processing (c).

**h) High-level syntax:** Above the coding layer, many of the high-level syntax features of AVC have been retained or extended.Parameter sets contain information that can be shared for the decoding of several pictures or regions of the decoded video. The parameter set structure provides a robust mechanism for conveying data that are essential to the decoding process. Each syntax structure is placed into a logical data packet called a *network abstraction layer* (NAL) unit. Depending on the content of a two-byte NAL unit header, it is possible to readily identify the purpose of the associated payload data, e.g. parameter sets, data for decoding random accessible pictures, etc. Further elements of high-level syntax are specifications of access structures, management of coded and decoded picture buffers, signaling of video usability information (VUI) and supplemental enhancement information (SEI).

**i) Extended format and quality ranges:** Version 1 of HEVC allowed only using 4:2:0 color sampling, with components represented by a maximum bit depth of 10. Version 2 enables using other color sampling schemes such as 4:2:2 and 4:4:4, and allows representing up to 16 bit video. In order to achieve this efficiently in the range of high bit rates, also some elements of entropy coding are modified when used in this context. Further tools are also added for more efficient lossless and near-lossless coding, whereas version 1 only supported simple lossless coding of CUs by invoking bypass of transform, quantization and loop filtering mechanisms.

**j) Multi-view and scalable coding:** Version 2 introduces more sophisticated layering mechanisms in the high-level syntax, allowing to establish hierarchical bitstream structures also associated with dependency structures between parts of information associated with the layers, as well as the related access mechanism. This can be used for the purpose of stereo/multi-view coding, where a decoder of a dependent view can refer to previously decoded pictures from another view (usually captured by a different camera) to save bit rate by utilizing inter-view redundancy; herein, motion compensation tools can be identically used to perform disparity compensated prediction from a reference picture captured at same time instance in a different view. Another usage is scalable coding, where an enhancement layer decoder can take reference to a lower layer, e.g. the same picture decoded by lower resolution or quality. In both multi-view and scalable coding existing version 1 decoder implementations can be re-used for reconstructing the pictures associated to the different layers; in the case of scalability with varying spatial resolution between the layers, filters for performing the re-sampling are specified for the additional inter-layer processing. As a specific type of layers, auxiliary pictures are furthermore defined, which could for example be used to represent alpha maps or depth maps.

# Profiles, levels and tiers

Similar to previous video compression standards, HEVC defines conformance points in terms of profiles (combinations of decoding tools and associated bitstream syntax that is expected to be interpreted by a decoder), and levels (typically maximum sizes of pictures and frame rates, maximum bit rate, buffer capacity etc.). Currently, a total of 13 levels, ranging in support for typical picture sizes as small as VGA at the low end, up to as large as 8Kx4K ultra HD at the high end, are specified. For more flexibility, HEVC also defines two tiers in some of the levels, basically defining two different limits in terms of the maximum number of bits per second that a conforming decoder should be able to process.

Version 1 contained three profiles (all supporting only 4:2:0 color sampling):

* The Main profile (8 bit);
* The Main 10 profile (10 bit, otherwise same as Main);
* The Main Still Picture profile (8 bit, only intra coding tools from Main).

The more recent version 2 additionally specifies a whole bundle of profiles, collectively referred to as the format range extensions profiles, which extend the applicability of the main and main still picture profile in various regards of color sampling and bit depth:

* The Monochrome, Monochrome 12 and Monochrome 16 profiles;
* The Main 12 profile;
* The Main 4:2:2 10 and Main 4:2:2 12 profiles;
* The Main 4:4:4, Main 4:4:4 10, and Main 4:4:4 12 profiles;
* The Main Intra, Main 10 Intra, Main 12 Intra, Main 4:2:2 10 Intra, Main 4:2:2 12 Intra, Main 4:4:4 Intra, Main 4:4:4 10 Intra, Main 4:4:4 12 Intra, and Main 4:4:4 16 Intra profiles;
* The Main 4:4:4 Still Picture and Main 4:4:4 16 Still Picture profiles;
* The High Throughput 4:4:4 16 Intra profile.

Version 2 furthermore contains

* The Multiview Main profile, enabling the layered dependent coding of multiple views (8 bit, 4:2:0 color sampling);
* The Scalable Main (8 bit) and Scalable Main 10 profiles (both with 4:2:0 color sampling over all layers).

# Performance

The JCT-VC carried out a formal subjective verification test of HEVC, where HEVC encoding was compared to its predecessor AVC. Subjective evaluation results for 5 test sequences are presented in Fig. 3, in the form of MOS vs. bit rate plots. The objective quality metric (PSNR) values for the same test points are also plotted on the same graph using a second vertical axis. Note that the scales for the two vertical axes in these plots are independently selected, and thus no direct correlation between subjective (i.e. MOS) and objective (i.e. PSNR) plots is demonstrated. In all plots, circles and triangles represent results for actual test points while the curves between them were calculated using cubic spline interpolation with the bit rate in a logarithmic scale, as in typical Bjøntegaard-Delta bit rate (BD-rate) computation [6]. Only the parts of the curves related to BD-rate calculation, either for MOS or PSNR BD-rate, are displayed. The solid and dotted lines represent MOS and PSNR curves, respectively. Confidence intervals are displayed for each MOS test point, where the error bars depict half-CI above and below the mean MOS value so that the full height of the error bar is the CI. The PSNR results presented are for the luma only.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  | Legend: |

Fig. 3. Subjective and objective evaluation results; subjective results include 95 % confidence limits

For all video sequences tested here (these were not included in the set of sequences that had been used during the development of the standard), more than a 50 % bit rate savings was observed – with an average of 60 % for the high quality range. The comparison of objective and subjective results shows that the difference between the objective and subjective bit rate savings for given test points is approximately 16 % on average across a wide bit rate range, i.e. the bit rate savings observed in subjective evaluations is typically 16 % larger than the savings indicated by measuring related PSNR values. This indicates that HEVC is improving the subjective quality compared to AVC even more than the objective measurements.

# Ongoing developments

Currently, two more extensions for HEVC are under development. One of them is nicknamed as “3D-HEVC”, which adds specific coding tools for efficiently exploiting characteristics of depth maps in their compression, and makes use of the redundancy between video pictures and the depth maps that are associated with them. For example, a motion vector that had already been coded/decoded for the video texture can be re-used for a collocated depth map, or the depth information can be used for improving the inter-view prediction. Representations including texture and depth information are expected to be needed in the context of autostereoscopic and multi-view displays, which need to use depth cues for synthesizing additional views. 3D-HEVC is planned to be finalized by early 2015.

The second extension under development defines specific tools that provide improved compression of computer generated content (also nicknamed as screen content), which is usually less noisy, has more uniformly colored areas, repetition of structures and cleaner edges than camera captured content. It is planned to be finalized by early 2016. Beyond the fact that computer generated content is becoming more and more prominent in the production of moving pictures, support for wireless displays is another foreseen application area.

# Resources

JCT-VC documents are publicly available from <http://phenix.it-sudparis.eu/jct/>, and the HM reference software repository is available at <http://hevc.hhi.fraunhofer.de/>.

# References

1. 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), May 2003 (and subsequent editions).
2. G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol*., vol. 22, Dec. 2012, pp. 1649–1668.
3. G. J. Sullivan, J. Boyce, Y. Chen, and J.-R. Ohm, “Standardized extensions of High Efficiency Video Coding (HEVC),” IEEE Journal of Selected Topics in Signal Processing, vol. 7, pp. 1001–1016, Dec. 2013
4. J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, “Comparison of the coding efficiency of video coding standards – including high efficiency video coding (HEVC),” *IEEE Trans. Circuits Syst. Video Technol*., vol. 22, Dec. 2012, pp. 1669–1684.
5. F. Bossen, B. Bross, K. Sühring, D. Flynn, “HEVC Complexity and Implementation Analysis,” *IEEE Trans. Circuits Syst. Video Technol*., vol. 22, Dec. 2012, pp xx-xx.
6. G. Bjøntegaard, "Improvements of the BD-PSNR model", ITU-T SG16/Q6, 35th VCEG Meeting, Berlin, Germany, Doc. VCEG-AI11, July 2008.