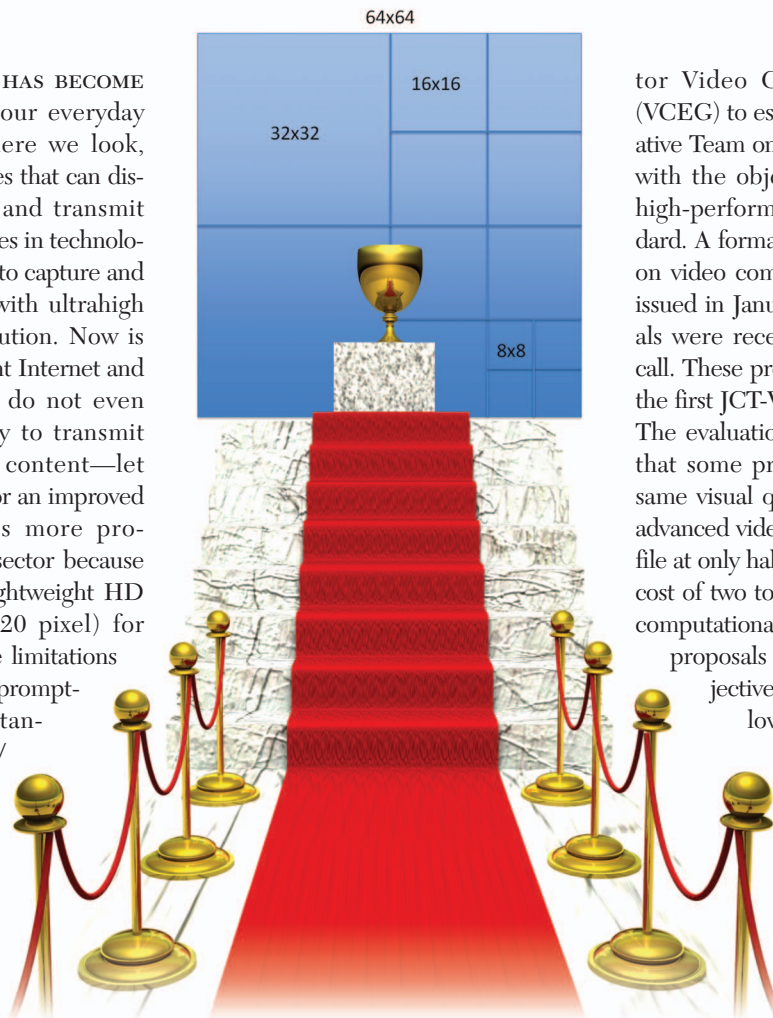


HEVC: The New Gold Standard for Video Compression

How does HEVC compare with H.264/AVC?

By Mahsa T. Pourazad, Colin Doutre, Maryam Azimi, and Panos Nasiopoulos

DIGITAL VIDEO HAS BECOME ubiquitous in our everyday lives; everywhere we look, there are devices that can display, capture, and transmit video. The recent advances in technology have made it possible to capture and display video material with ultrahigh definition (UHD) resolution. Now is the time when the current Internet and broadcasting networks do not even have sufficient capacity to transmit large amounts of HD content—let alone UHD. The need for an improved transmission system is more pronounced in the mobile sector because of the introduction of lightweight HD resolutions (such as 720 pixel) for mobile applications. The limitations of current technologies prompted the International Standards Organization/International Electrotechnical Commission Moving Picture Experts Group (MPEG) and International Telecommunication Union—Telecommunication Standardization Sec-



tor Video Coding Experts Group (VCEG) to establish the Joint Collaborative Team on Video Coding (JCT-VC), with the objective to develop a new high-performance video coding standard. A formal call for proposals (CfPs) on video compression technology was issued in January 2010, and 27 proposals were received in response to that call. These proposals were presented at the first JCT-VC meeting in April 2010. The evaluations that followed showed that some proposals could reach the same visual quality as H.264/MPEG-4 advanced video coding (AVC) high profile at only half of the bit rate and at the cost of two to ten times the increase in computational complexity. Some other proposals could achieve good subjective quality and bit rates with lower computational complexity than the reference AVC high-profile encoder. Since then, JCT-VC has put a considerable effort toward the development of a new compression standard known as the

high-efficiency video coding (HEVC) standard, with the aim to significantly improve the compression efficiency compared with the existing H.264/AVC high profile. The first task of the JCT-VC group was to integrate the key features of seven top high-performing proposals into a single test model under consideration (TMuC), which became the basis for the first HEVC software codec known as HM [1]. Since then, JCT-VC has held several meetings and reviewed hundreds of contributions received from industry and academia. These submissions have been carefully evaluated, and the best ones were included in the HEVC standard.

Some of the key elements of the current version of the HEVC test model (TMuC 5.0) are as follows: 1) a more flexible block structure, with block sizes ranging from 64×64 down to 8×8 pixels using recursive quad-tree partitioning, 2) improved mechanisms to support parallel encoding and decoding, including tiles and wavefront parallel processing (WPP), 3) more intraprediction modes (35 in total, most of which are directional), which can be done at several block sizes, 4) support for several integer transforms, ranging from 32×32 down to 4×4 pixels, as well as non-square transforms, 5) improved motion information encoding, including a new merge mode, where just an index indicating a previous block is signaled in the bit stream, and 6) extensive in-loop processing on reconstructed pictures, including a deblocking filter, sample adaptive offset (SAO), and adaptive loop filtering (ALF). This article provides an overview of HEVC as it currently stands, discusses its key

One of the major contributors to the higher compression performance of HEVC is the introduction of larger block structures with flexible subpartitioning mechanisms.

features, and compares its performance with the H.264/MPEG-4 AVC high-profile standard.

HEVC DESIGN

Although HEVC has not yet been finalized, the key elements of this new standard have been identified. In this section, we review the current design of HEVC and discuss the features that differentiate it from H.264/AVC. HEVC is still being fine-tuned, and it will include other features by the time it reaches its final form. It is important to note that this article serves as a snapshot of the current status of HEVC as it gets close to its completion status. In that respect, the final version will differ somewhat from what is described.

Figure 1 shows the block diagram of the basic HEVC design—as it is implemented in the HM 5.0 software codec. As can be observed, the main structure of the HEVC

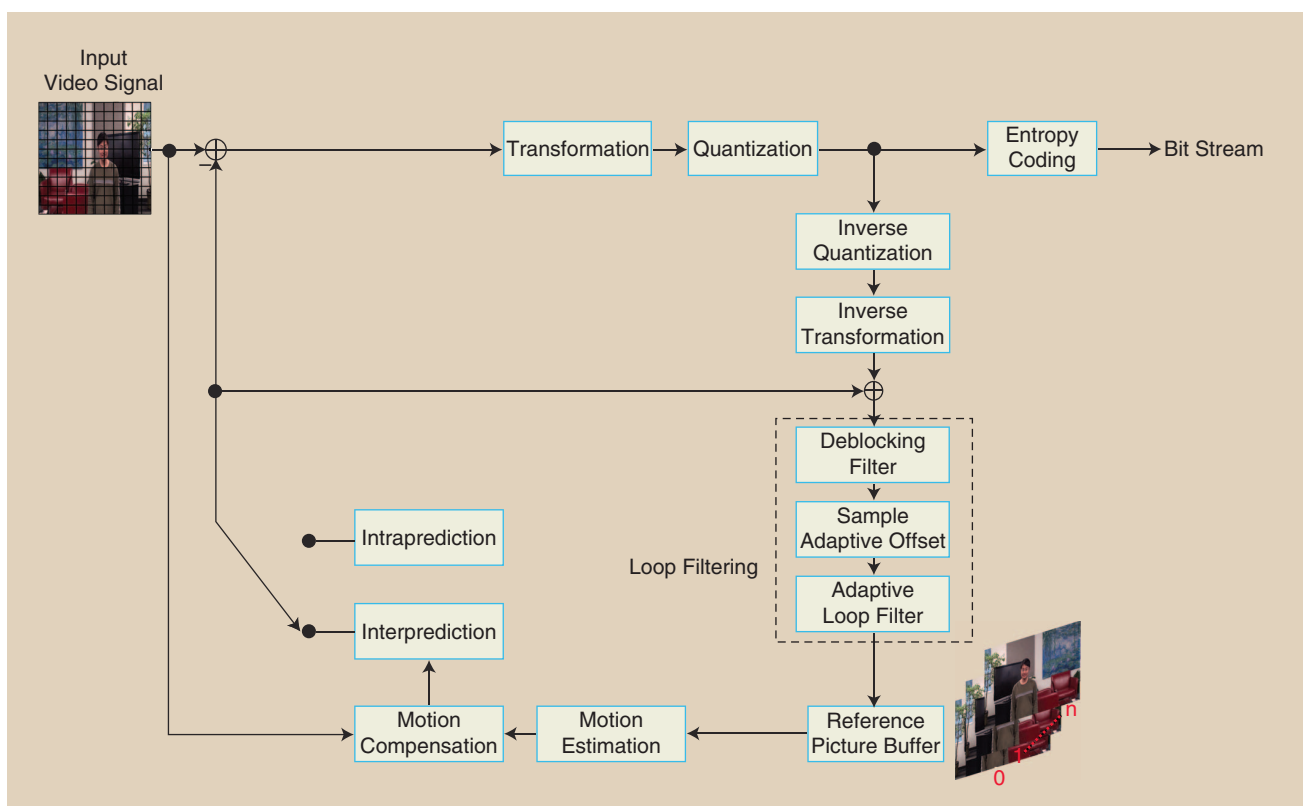


FIGURE 1. Block diagram of an HM-5.0 encoder.

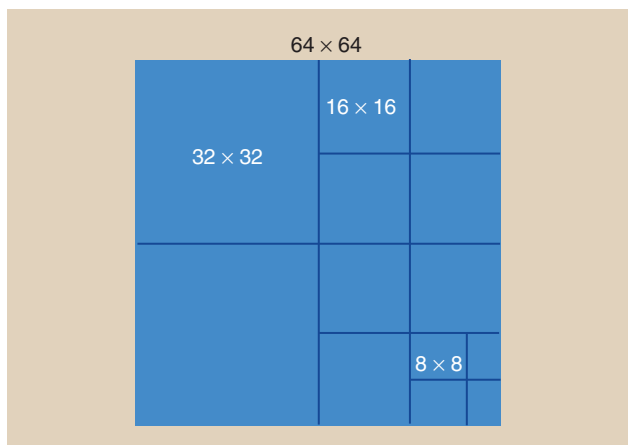


FIGURE 2. Partitioning of a 64×64 LCU to various sizes of CU.

encoder resembles that of the H.264/AVC. So far, only one profile (main) has been specified for HEVC, and more profiles and a number of levels are being considered. The key features of the latest version of HEVC are described in detail in the following sections.

PICTURE PARTITIONING

Similar to the conventional video coding standards, HEVC is a block-based hybrid-coding scheme. One of the major contributors to the higher compression performance of HEVC is the introduction of larger block structures with flexible subpartitioning mechanisms. The basic block in HEVC is known as the largest coding unit (LCU) and can be recursively split into smaller coding units (CUs), which in turn can be split into small prediction units (PUs) and transform units (TU). These concepts are explained in the following subsections.

CODING UNITS

In H.264/AVC, each picture is partitioned into 16×16 macroblocks, and each macroblock can be further split into smaller blocks (as small as 4×4) for prediction [2]. Other standards, such as MPEG-2 and H.263, were more rigid regarding block sizes for motion compensation and transforms. Such a rigid structure may not perform well for all kinds of content; large blocks will generally work better for smooth regions of a picture, whereas edges and texture regions will often benefit from smaller block sizes. As the picture resolution of videos increases from standard definition to HD and beyond, the chances are that the picture will contain larger smooth regions, which can be encoded more effectively when large block sizes are used. This is the reason that HEVC supports larger encoding blocks than H.264/AVC, while it also has a more flexible partitioning structure to allow smaller blocks to be used for more textured and—in general—uneven regions.

In HEVC, each picture is partitioned into square picture areas called LCUs that can be as large as 64×64 . The LCU notion in HEVC is generally similar to that of a macroblock in the previous coding standards. LCUs can be further split into smaller units called CUs, which are used as the basic unit for intra- and intercoding. The size of CUs can be as large as that of LCUs or can be recursively split into four equally sized CUs and become as small as 8×8 , depending on the picture content. Because of recursive quarter-size splitting, a content-adaptive coding tree structure comprised of CUs is created in HEVC [3]. Figure 2 shows an example of partitioning a 64×64 LCU to various sizes of CUs.

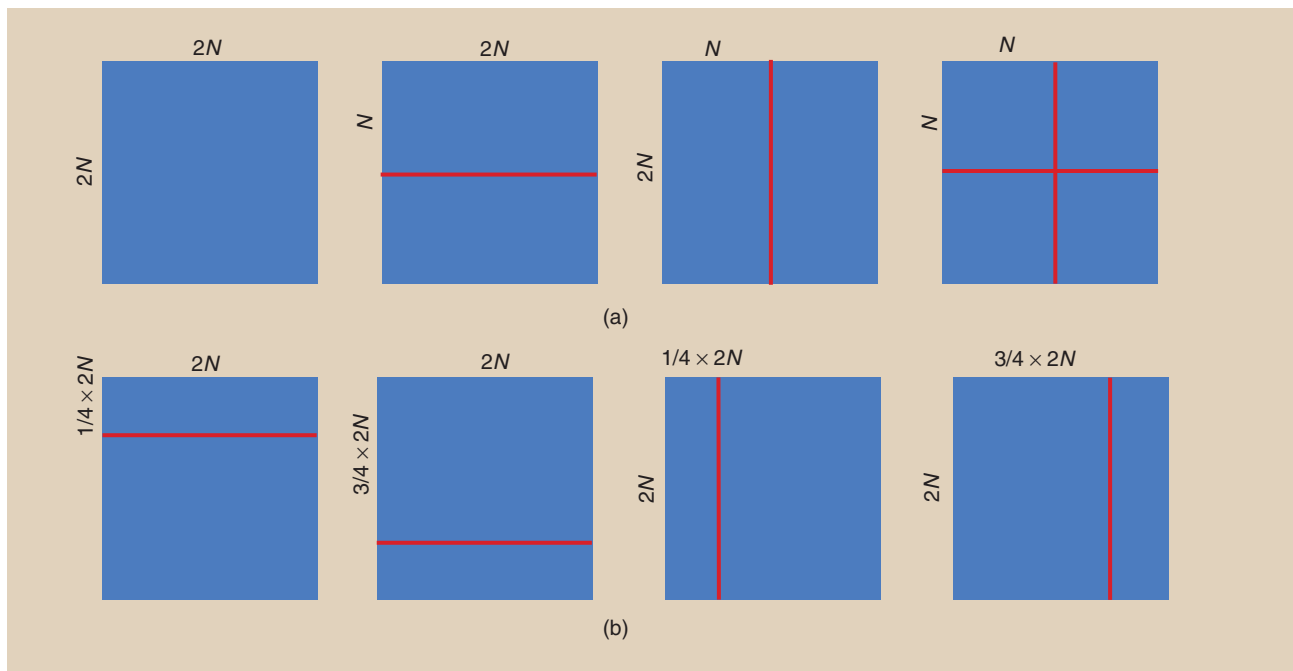


FIGURE 3. (a) Symmetric and (b) asymmetric PUs.

PREDICTION UNITS

Each CU can be further split into smaller units, which form the basis for prediction. These units are called PUs. Each CU may contain one or more PUs, and each PU can be as large as their root CU or as small as 4×4 in luma block sizes. While an LCU can recursively split into smaller and smaller CUs, the splitting of a CU into PUs is nonrecursive (it can be done only once). PUs can be symmetric or asymmetric. Symmetric PUs can be square or rectangular (nonsquare) and are used in both intraprediction (uses only square PUs) and interprediction. In particular, a CU of size $2N \times 2N$ can be split into two symmetric PUs of size $N \times 2N$ or $2N \times N$ or four PUs of size $N \times N$. Asymmetric PUs are used only for interprediction. This allows partitioning, which matches the boundaries of the objects in the picture [3]. Figure 3 shows partitioning of a CU to symmetric and asymmetric PUs.

TRANSFORM UNITS

Similar to previous video-coding standards, HEVC applies a discrete cosine transform (DCT)-like transformation to the residuals to decorrelated data. In HEVC, a transform unit (TU) is the basic unit for the transform and quantization processes. The size and the shape of the TU depend on the size of the PU. The size of square-shape TUs can be as small as 4×4 or as large as 32×32 . Nonsquare TUs can have sizes of 32×8 , 8×32 , 16×4 , or 4×16 luma sam-

ples. Each CU may contain one or more TUs; each square CU may split into smaller TUs in a quad-tree segmentation structure. Figure 4 shows an example of how multiple TUs are arranged in an LCU, and Figure 5 illustrates an example for partitioning a 32×32 CU into PUs and TUs.

SLICES AND TILE STRUCTURE

The H.264/AVC video-coding standard used slices to support parallel encoding/decoding and also offer error resiliency. Each slice can be independently decoded, that is, without using information from any other slice. H.264/AVC introduced flexible macroblock ordering (FMO) as a tool for arranging macroblocks into slices in a highly flexible manner. While FMO could improve error resilience in high-loss situations, it was not widely used in practice because of the increased complexity and lower coding efficiency caused from disabling prediction across slice boundaries.

HEVC introduced tiles as a means to support parallel processing, with more flexibility than normal slices in H.264/AVC but considerably lower complexity than FMO. Tiles are specified by vertical and horizontal boundaries with intersections that partition a picture into rectangular regions [4]. Figure 6 shows an example of tile partitions that contain slices. The spacing of the row and column boundaries of tiles need not be uniform. This offers greater flexibility and can be useful for error resilience applications. In each tile, LCUs are

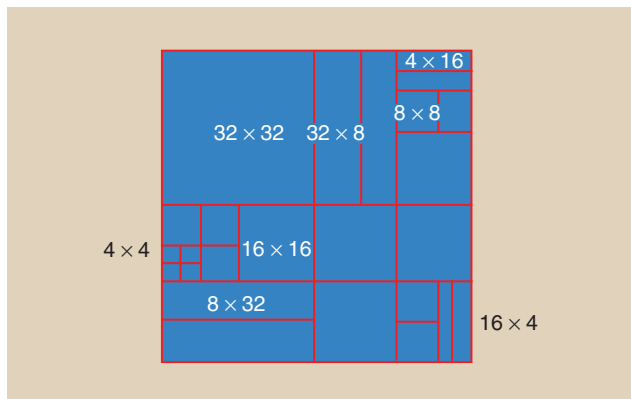


FIGURE 4. An example of arranging TUs in an LCU.

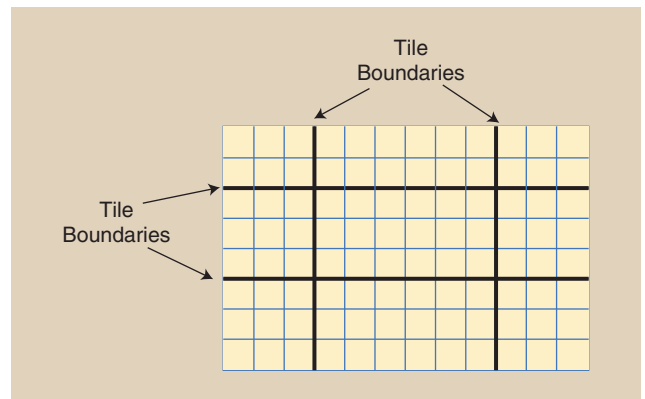


FIGURE 6. An example showing a picture partitioned into nine tiles.

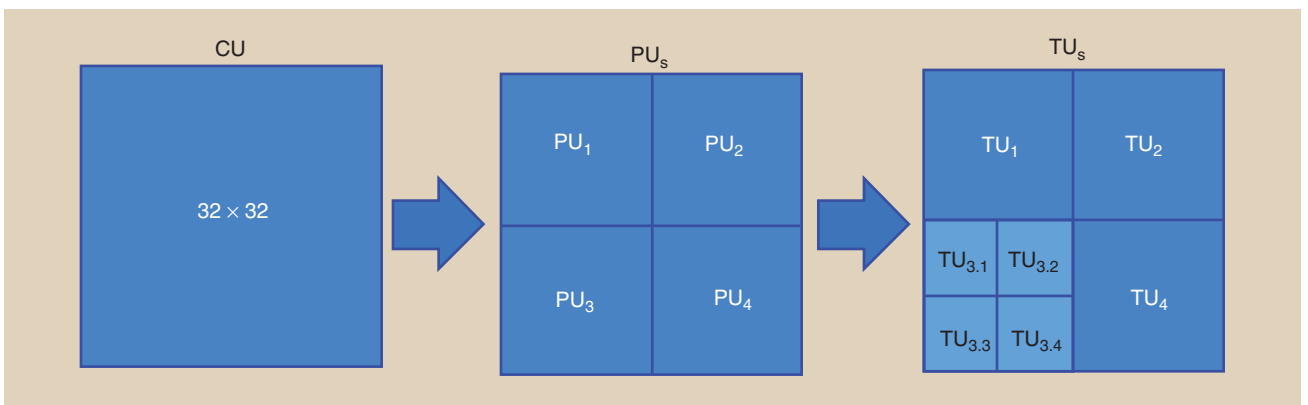


FIGURE 5. An example of partitioning a 32×32 CU to PUs and TUs.

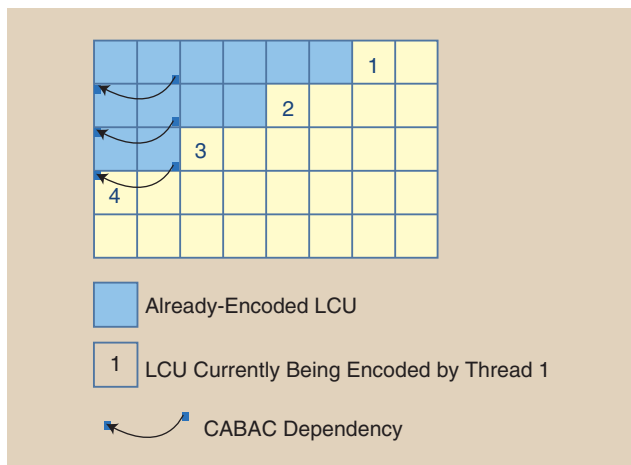


FIGURE 7. Illustration of WPP with four threads.

processed in a raster scan order. Similarly, the tiles themselves are processed in a raster scan order within a picture.

HEVC also supports slices, similar to slices found in H.264/AVC, but without FMO. Slices and tiles may be used together within the same picture. To support parallel processing, each slice in HEVC can be subdivided into smaller slices called entropy slices. Each entropy slice can be independently entropy decoded without reference to other entropy slices. Therefore, each core of a CPU can handle an entropy-decoding process in parallel [5].

WAVEFRONT PARALLEL PROCESSING

Slices and tiles provide mechanisms for parallel encoding and decoding in HEVC. However, they both come with a performance penalty since prediction dependencies are broken across boundaries and the statistics used in entropy

coding have to be initialized for every slice/tile. To avoid these problems, WPP is supported in HEVC [6]. Wavefront processing is a way to achieve parallel encoding and decoding without breaking prediction dependencies and using as much context as possible in entropy encoding. The basic concept is to start processing (either encoding or decoding) a new row of LCUs with a new parallel process (usually, a new thread) as soon as two LCUs have been processed in the row above (Figure 7). Two LCUs are required because intraprediction and motion vector prediction can depend upon data from both the LCU directly above the current one and the one above the right. The entropy-coding parameters are initialized based on the information obtained from the two fully encoded LCUs in the row above, which allows using as much context as possible in the new encoding thread.

INTRAFRAME CODING

Like H.264/AVC, HEVC uses block-based intraprediction to take advantage of spatial correlation within a picture. HEVC follows the basic idea of H.264/AVC intraprediction but makes it far more flexible. HEVC has 35 luma intraprediction modes compared with nine in H.264/AVC. Furthermore, intraprediction can be done at different block sizes, ranging from 4×4 to 64×64 (whatever size the PU has). Figure 8 shows the luma intraprediction modes of HEVC versus those of H.264/AVC. The number of supported prediction modes varies based on the PU size (see Table 1) [3]. HEVC also includes a planar intraprediction mode, which is useful for predicting smooth picture regions. In planar mode, the prediction is generated from the average of two linear interpolations (horizontal and vertical); see [7] for details.

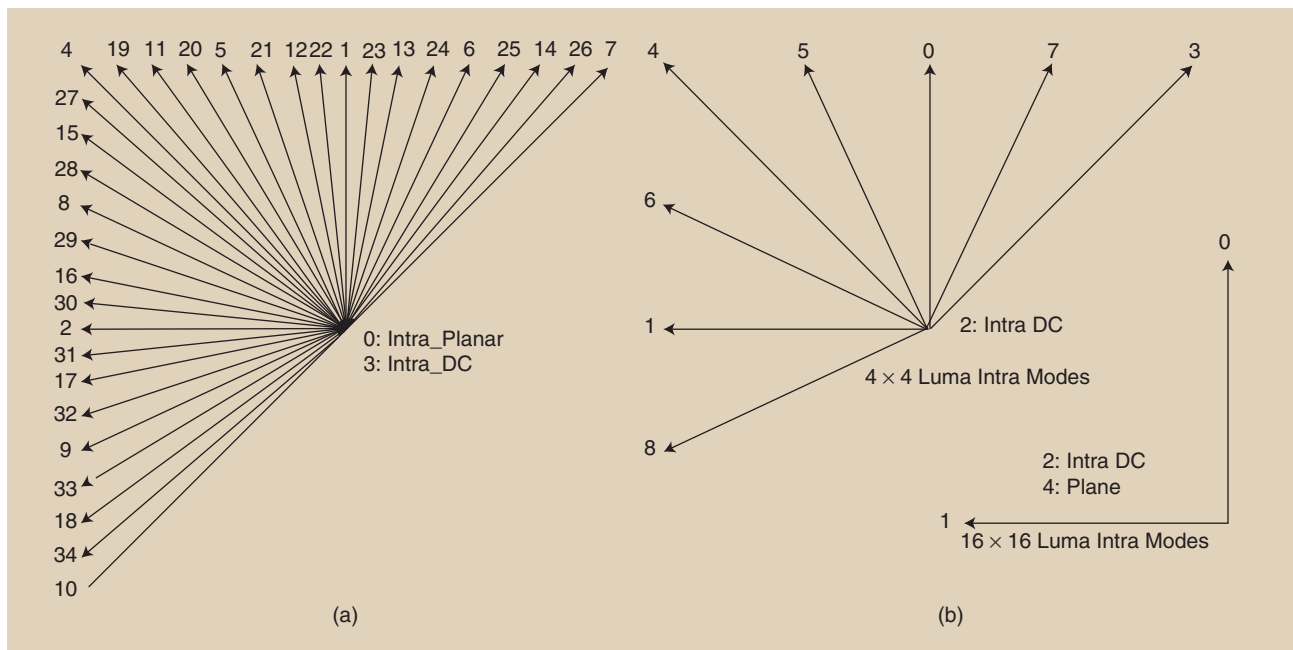


FIGURE 8. Luma intraprediction modes of (a) HEVC and (b) H.264/AVC.

To improve the performance of intraprediction, mode-dependent intrasmoothing (MDIS) is used for some intramodes. MDIS involves applying a simple low-pass finite impulse response filter with coefficients (1, 2, 1)/4 to the samples being used for prediction. This smoothing of the reference signal improves the prediction performance, especially, for large PUs. MDIS is enabled based on the PU size and intramode. In general, MDIS is used for large PU sizes and directional modes, except horizontal and vertical modes (see Figure 9 for an example of an MDIS application).

The number of chroma intraprediction modes in HEVC is also more than H.264/AVC. While H.264/AVC supports four chroma intraprediction modes, HEVC has six chroma intraprediction modes: direct mode (DM), linear mode (LM), vertical (mode 0), horizontal (mode 1), DC (mode 2), and planar (mode 3). In principle, DM and LM exploit the correlation of the luma component and chroma component [8]. DM is selected if the texture directionality of luma and chroma components look similar. In this case, it uses exactly the same mode as that of the luma component. On the other hand, LM is selected if the sample intensities of luma and chroma components are highly correlated. In this case, the chroma components are predicted using luma components reconstructed by the linear model relationship. Because the type of correlations exploited by DM and LM are different, the two modes complement each other in terms of coding performance. Because of the existing correlation between luma and chroma, DM and LM are the most frequently selected modes for intraprediction of the chroma component [9].

INTERPREDICTION

Intercoded pictures are those coded with reference to other pictures. Interprediction takes advantage of the similarities of each picture with its temporal neighbors and exploits these similarities. The enhancements of interprediction introduced in HEVC, compared with H.264/AVC, are described later.

VARIABLE PU SIZE MOTION COMPENSATION

As described earlier, each LCU can be recursively split into smaller square CUs, and these CUs can be split once more into smaller PUs, which can be square or rectangular (non-square). Each PU, coded using interprediction, has a set of

Table 1. Luma intraprediction modes supported for different PU sizes.

| PU Size | Intraprediction Modes |
|---------|-----------------------|
| 4 × 4 | 0–16, 34 |
| 8 × 8 | 0–34 |
| 16 × 16 | 0–34 |
| 32 × 32 | 0–34 |
| 64 × 64 | 0–2, 34 |

motion parameters, which consists of a motion vector, a reference picture index, and a reference list flag.

Inter-coded CUs can use symmetric and asymmetric motion partitions (AMPs). AMPs allow for asymmetrical splitting of a CU into smaller PUs. Figure 10 shows an example where a 32 × 32 CU is asymmetrical partitioned. AMP can be used on CUs of size 64 × 64 down to 16 × 16. AMP improves the coding efficiency since it allows PUs to more accurately conform to the shape of objects in the picture without requiring further splitting [10].

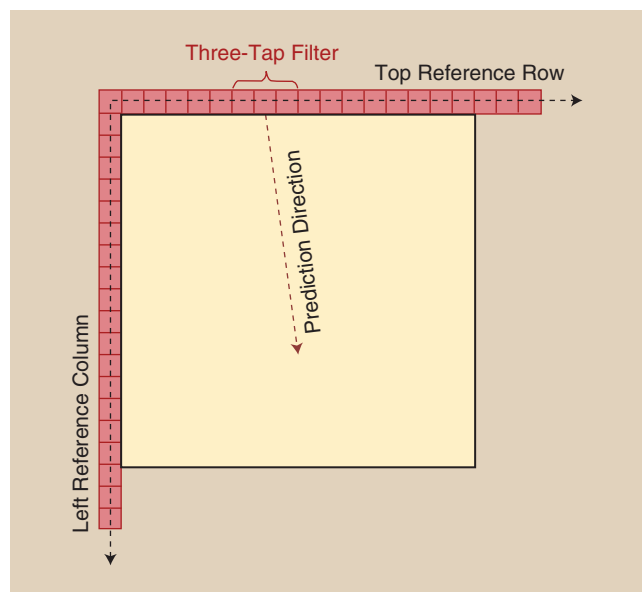


FIGURE 9. An example of using the MDIS filter for intraprediction.

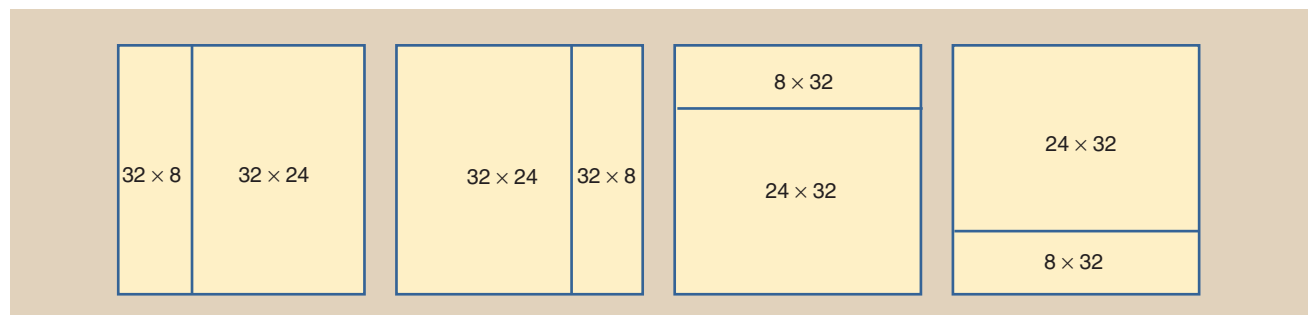


FIGURE 10. Asymmetric motion partitions of a 32 × 32 CU.

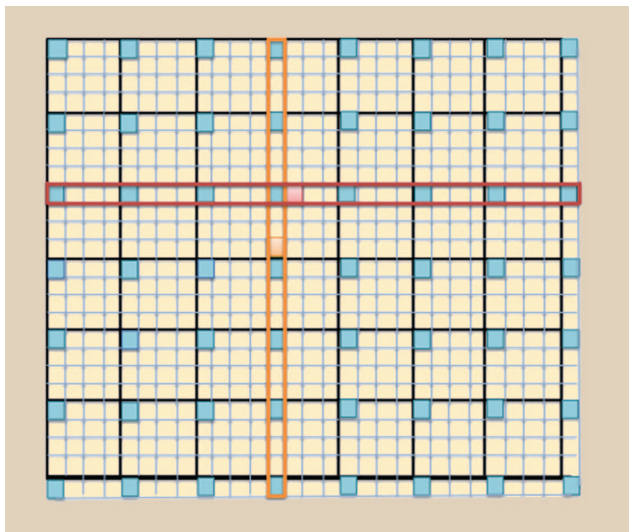


FIGURE 11. Sub-pel interpolation from full-pel samples.

IMPROVED SUBPIXEL INTERPOLATION

Like H.264/AVC, the accuracy of motion compensation in HEVC is 1/4 pel for luma samples. To obtain the noninteger luma samples, separable one-dimensional eight-tap and seven-tap interpolation filters are applied horizontally and vertically to generate luma half-pel and quarter-pel samples, respectively [11]. An illustration of the surrounding full-pel samples used in generating the fractional-pel values is provided in Figure 11, and the filter coefficients for each noninteger luma position are available in Table 2. Note that, unlike H.264/AVC, the quarter-pel values are calculated from the integer luma samples with a longer filter, instead of using bilinear interpolation on the neighboring half-pel and integer-pel values.

The prediction values for chroma components are similarly generated by applying a one-dimensional four-tap DCT-based interpolation filter. The accuracy for chroma sample prediction is one eighth of chroma samples.

MOTION PARAMETER ENCODING AND IMPROVED SKIP MODE

In H.264/AVC, motion vectors (MVs) are encoded by calculating a predicted motion vector and encoding the difference between the desired MV and the predicted one. The predicted MV is formed as the median of three surrounding MVs (left, above, and above right). Furthermore, H.264/AVC has a SKIP mode, where no motion parameters or

quantized residuals are encoded in the bit stream, but instead, the motion parameters are inferred from a colocated MB in the previous frame.

In HEVC, MVs can be predicted either spatially or temporally. Furthermore, HEVC introduces a technique called motion merge [1]. For every intercoded PU, the encoder can choose between 1) using explicit encoding of motion parameters (i.e., using motion vector prediction and encoding the MV difference and reference picture), 2) using motion merge mode, or 3) using the improved SKIP mode.

Motion merge mode involves creating a list of previously coded neighboring PUs (called candidates) for the PU being encoded. The candidates are either spatially or temporally close to the current PU. The encoder signals, which candidate from this motion merge list, will be used, and the motion information for the current PU is copied from the selected candidate. Note that motion merge avoids the need to encode a motion vector for the PU; instead, only the index of a candidate in the motion merge list is encoded.

In the new SKIP mode in HEVC, the encoder also encodes the index of a motion merge candidate, and the motion parameters for the current PU are copied from the selected candidate. This allows areas of the picture that change very little between frames or have constant motion to be encoded using very few bits.

TRANSFORM AND QUANTIZATION

Similar to H.264/AVC, HEVC applies a DCT-like integer transform on the prediction residual. HEVC includes transforms that can be applied to blocks of sizes ranging from 4×4 to 32×32 pixels. The basis vectors of the 4×4 , 8×8 , 16×16 , and 32×32 transforms are available in [12]. HEVC also supports transforms on rectangular (non-square) blocks, where the row transform and column transforms have different sizes. The integer transforms used in HEVC are better approximations of the DCT than the transforms used in H.264/AVC. The basis vectors of the HEVC transforms have equal energy, so there is no need to compensate for the different norms, as in H.264/AVC.

HEVC also incorporates a 4×4 discrete sine transform (DST), which is used for blocks coded with some directional intraprediction modes. When using intraprediction, the pixels close to the ones used for prediction (i.e., near the top or left boundaries) will usually be predicted more accurately than the pixels further away. Therefore, the residuals tend to be larger for pixels away from the boundaries. The DST will usually be better at encoding these kinds of residuals, because the DST basis functions start low and increase, compared with the DCT basis functions that start high and decrease [13].

ENTROPY CODING

After transformation, entropy coding is applied to code all the syntax elements and quantized transform coefficients. In H.264/AVC, context-adaptive variable-length coding (CAVLC) is the base entropy coder, and context-adaptive

Table 2. Luma Sub-pel interpolation filter coefficients.

| Position | Filter Coefficients |
|----------|----------------------------------|
| 1/4 | {-1, 4, -10, 58, 17, -5, 1, 0} |
| 2/4 | {-1, 4, -11, 40, 40, -11, 4, -1} |
| 3/4 | {1, -5, 17, 58, -10, 4, -1, 0} |

binary arithmetic coding (CABAC) is optionally used in the main and high profiles. CABAC can provide better coding efficiency than CAVLC because of its arithmetic coding engine and more sophisticated context modeling. While CABAC improves the coding efficiency, it increases coding complexity. This is more pronounced at higher bit rates [small quantization parameters (QPs)], where the transform coefficient data have a dominant role in encoded bit streams. In HEVC, to improve the worst-case throughput, the codec uses a higher-throughput alternative mode for coding transform coefficient data. Figure 12 illustrates the block diagram of HEVC entropy coding. As can be seen, there are two modes of HEVC entropy coding: high-efficiency binarization (HEB) and high-throughput binarization (HTB) [14]. The HEB mode is entirely CABAC based while the HTB mode is partially based on the well-known CAVLC residual coding module. HTB is intended to serve as the high-throughput mode of HEVC, and its use is signaled at slice level (one bit identifier indicating whether HTB is used). In HTB mode, all syntax elements except the residual coefficients are coded using CABAC while the residual coefficients are coded using CAVLC. Using this harmonized design, HEVC entropy coding uses the best features of both CABAC and CAVLC coding (i.e., high efficiency and low complexity, respectively).

LOOP FILTERING

Referring to Figure 1, loop filtering is applied after inverse quantization and transform, but before the reconstructed picture is used for predicting other pictures through motion compensation. The name loop filtering reflects the fact that filtering is done as part of the prediction loop rather than postprocessing. H.264/AVC includes an in-loop deblocking filter. HEVC employs a deblocking filter similar to the one used in H.264/AVC but also expands an in-loop processing by introducing two new tools: SAO and ALF. These techniques are intended to undo the distortion introduced in the main steps of the encoding process (prediction, transform, and quantization). By including filtering as part of the prediction loop, the pictures will serve as better references for motion-compensated prediction since they have less encoding distortion.

DEBLOCKING FILTER

Blocking is known as one of the most visible and objectionable artifacts of block-based compression methods. For this reason, in H.264/AVC, low-pass filters are adaptively applied to block boundaries according to the boundary strength. This improves the subjective and objective quality of the video.

HEVC uses an in-loop deblocking filter similar to the one used in H.264/AVC. In HEVC, there are several kinds of block boundaries, such as CUs, PUs, and TUs). The

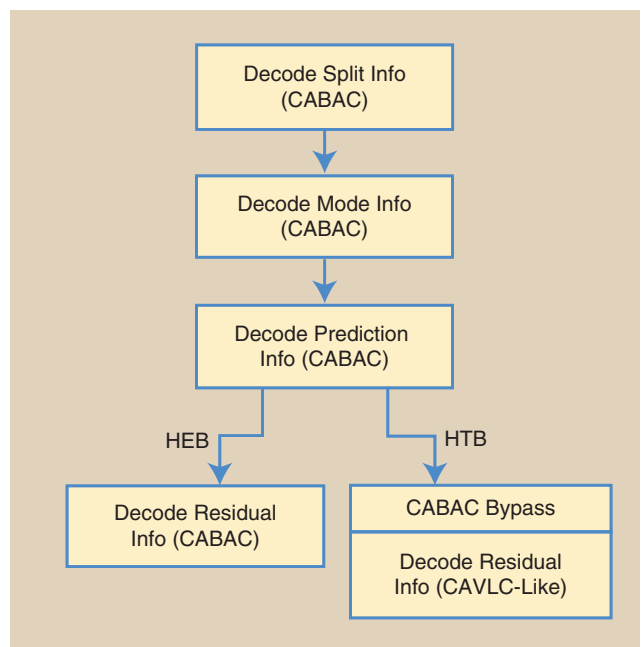


FIGURE 12. HEVC entropy coding.

set of boundaries that may be filtered in HEVC is the union of all of these boundaries (except for 4×4 blocks, which are not filtered to reduce complexity). For each boundary, a decision is made to turn the deblocking on or off and whether to apply strong or weak filtering. This decision is based on the pixel gradients across the boundary and thresholds derived based on the QP in the blocks. For more details on the deblocking filter, refer to [1].

SAMPLE ADAPTIVE OFFSET

SAO [15] is a new coding tool introduced in HEVC, which involves classifying pixels into different categories and adding a simple offset value to each pixel based on its category. SAO classifies reconstructed pixels into different categories based on either intensity or edge properties. It then adds an offset, either band offset (BO) or edge offset (EO), to the pixels in each category in a region to reduce distortion.

BO classifies all pixels of a region into multiple bands, with each band containing pixels in the same intensity interval. The intensity range is divided into 32 equal intervals from zero to the maximum intensity. For example, for 8-b data, the maximum value is 255, so the bands will be $256/32 = 8$ pixels wide. The 32 bands are divided into two groups. One group consists of the central 16 bands while the other group consists of the rest 16 bands (see the example in Figure 13). The

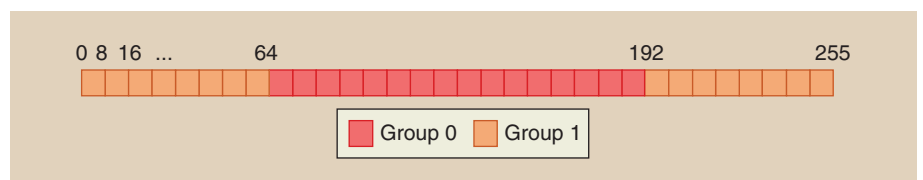


FIGURE 13. An example of intensity bands and groups of bands in BO mode, for 8 b.

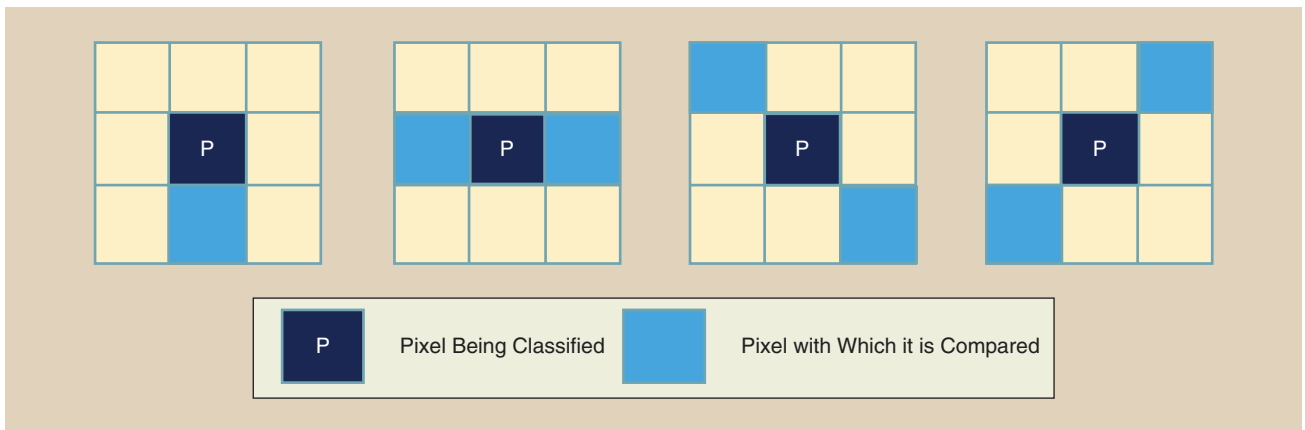


FIGURE 14. Patterns used in EO mode.

encoder decides which group of bands to apply SAO, so 16 offsets will be encoded in the bit stream [15].

EO uses one of four one-dimensional three-pixel patterns to classify pixels based on their edge direction, as illustrated in Figure 14. Each pixel can be classified as a peak (if it is greater than two neighbors), valley (if it is less than the two neighbors), edge (if it is equal to one neighbor, categories 2 and 3), or none of these. Four offset values will be calculated for these four categories.

The encoder can choose to apply either BO or EO to different regions of a picture. It can also signal that neither BO nor EO is used for a region.

ADAPTIVE LOOP FILTERING

In HEVC, an ALF is applied to the reconstructed signal after the deblocking filter and SAO. The filter is adaptive in the sense that the coefficients are signaled in the bit stream and can therefore be designed based on image content and distortion of the reconstructed picture. The filter is used to restore the reconstructed picture such that the mean-squared error between the source picture and the reconstructed picture is minimized.

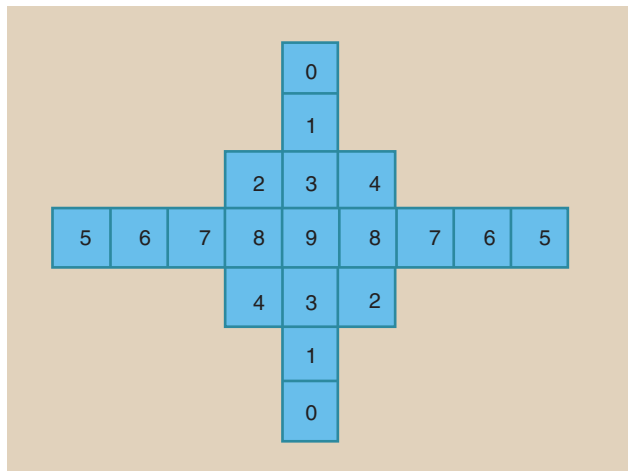


FIGURE 15. An ALF shape.

The current HEVC draft [3] uses a single filter shape, a cross overlaid on a 3×3 square with nine coefficients to be encoded in the bit stream (Figure 15) [16]. Note that the number of taps in the filter is greater than nine due to symmetry.

There are two modes that can be used for applying different filters to different pixels within each picture: region-based adaptation (RA) and block-based adaptation (BA). In RA mode, the picture is divided into 16 regions of equal size. These regions can be merged, and each region remaining after merging will have its own filter (with a unique set of coefficients). In BA mode, 4×4 blocks are classified into 1 of 16 categories based on edge activity and direction. These categories can be merged, and in the end, one filter will be designed for each of the categories left after merging. The filter coefficients for each region can be calculated based on the autocorrelation and cross-correlation of the original pixels and the reconstruction pixels in the region (using Wiener-Hopf equations) [17].

The ALF can be enabled or disabled for different picture areas based on the partitioning of LCUs into CUs (in a quad-tree segmentation structure).

PERFORMANCE COMPARED TO H.264/AVC

We evaluated the performance of the current HEVC model (HM 5.1 software) and compared it with that of the high-profile H.264/AVC standard (JM16.2 software). Four test sequences with different resolution and frame rates are selected from the database provided to MPEG for CFP on HEVC (see Table 3) [18]. All the test videos are in YUV 4:2:0 format and progressive. The configuration of H.264/AVC was as follows: high profile, hierarchical B pictures, group of pictures (GOP) length 8, CABAC entropy coding, and rate-distortion optimized quantization (RDOQ) enabled. These settings were recommended for comparing H.264/AVC to HEVC by MPEG/VCEG in the Joint CfPs (for more details, check the Alpha anchor in [18]).

For testing HEVC, the random access high efficiency (RA-HE) configuration was used [19], to ensure achieving the highest compression performance. The RA-HE

configuration is as follows: hierarchical B pictures, GOP length 8, ALF, SAO, and RDOQ were enabled (see [3] and [21] for more details). The QPs used were 26, 32, 37, and 44.

Figure 16 shows the rate-distortion (RD) curves for all the test sequences, and Table 4 lists the average PSNR improvement and average PSNR savings achieved by HEVC over the H.264/AVC standard. As it can be observed, the current HEVC design outperforms H.264/AVC by 29.14–45.54% in terms of bit rate or 1.4–1.87 dB in terms of PSNR. Subjective comparison of the quality of compressed videos—for the same (linearly interpolated) mean opinion score points—shows that HEVC outperforms H.264/AVC, yielding average bit-rate savings of 58% [20]. Note that, in the subjective tests performed by [20], the opinion of the viewers about the quality of compressed videos at different bit rates has been taken into account as the measure of quality, while in our objective tests, PSNR is used as the measure of quality. These objective and subjective results confirm that the goal of developing a high-efficiency video coding standard, which delivers the same visual quality as H.264/MPEG-4 AVC high profile at only half of the bit rate, has been accomplished.

Table 3. Test sequences.

| Sequences | Resolution | Frame Rate (fps) |
|------------------|---------------|------------------|
| People on street | 2,560 × 1,600 | 30 |
| Basketball drive | 1,920 × 1,080 | 50 |
| Race horses | 832 × 480 | 30 |
| Blowing bubbles | 416 × 240 | 50 |

Table 4. Average PSNR improvement and average bit rate saving achieved by HEVC (HM 5.1).

| Sequences | Average PSNR Improvement (dB) | Average Bit Rate Saving (%) |
|------------------|-------------------------------|-----------------------------|
| People on street | 1.87 | 31.86 |
| Basketball drive | 1.73 | 45.54 |
| Race horses | 1.84 | 39.89 |
| Blowing bubbles | 1.4 | 29.14 |

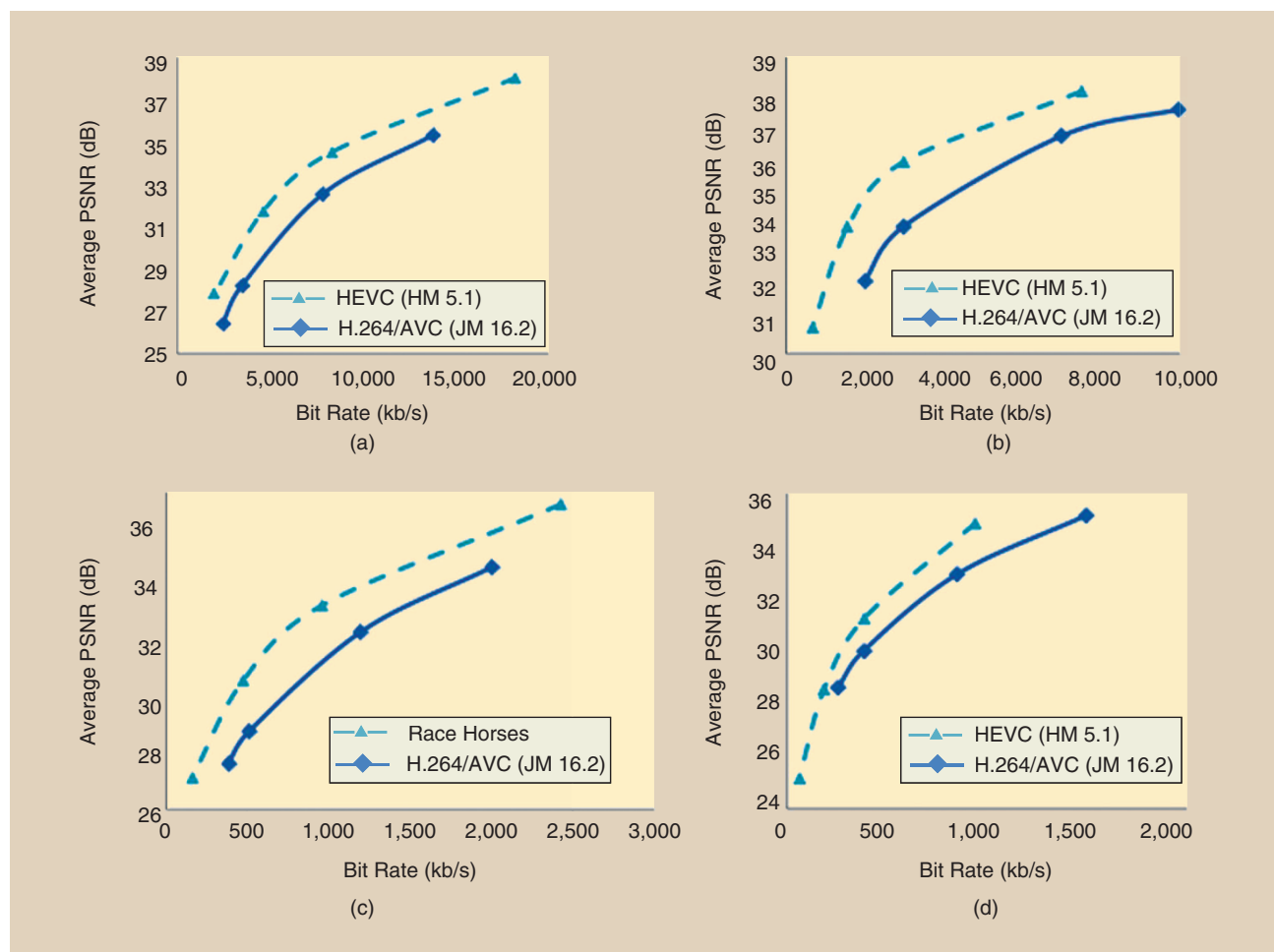


FIGURE 16. Comparison of HEVC and H.264/AVC coding efficiency. (a) People on the street, (b) basketball drive, (c) race horses, and (d) blowing bubbles.



Objective and subjective results confirm that the goal of developing a high-efficiency video coding standard, which delivers the same visual quality as H.264/MPEG-4 AVC high profile at only half of the bit rate, has been accomplished.

ABOUT THE AUTHORS

Mahsa T. Pourazad received her B.A.Sc. degree in electrical engineering from Iran University of Science and Technology in 2000, her M.A.Sc. degree in electrical and computer engineering from the University of Manitoba in 2004, and her Ph.D. degree in electrical and computer engineering from the University of British Columbia (UBC) in 2010. She is currently a research scientist at TELUS Communications Inc. and a system consultant at UBC (ICICS). Her research interests include three-dimensional (3-D) video processing, 3-D quality of experience, and multiview video compression. She has been an active Member of IEEE, the Standards Council of Canada (SCC), and MPEG.

Colin Doutré received his B.Sc. degree in electrical engineering from Queen's University, Kingston, Ontario, Canada, in 2005, and his M.A.Sc. and Ph.D. degrees in electrical and computer engineering from UBC in 2007 and 2012, respectively. His research interests include multiview video processing, 3-D video, and video compression. He has received numerous awards and scholarships, including the Governor General's Gold Medal for being selected as the top research-based master's student graduating from UBC in 2007.

Maryam Azimi received her B.E. degree in computer engineering from Ferdowsi University of Mashhad, Iran, in 2009. She is currently an M.A.Sc. student in electrical and computer engineering at the Electrical and Computer Engineering Department of UBC. Her research focuses on compression algorithms for stereoscopic video and high dynamic image range images/video.

Panos Nasiopoulos received his B.S. degree in physics from the Aristotle University of Thessaloniki, Greece, in 1980, and his B.S., M.S., and Ph.D. degrees in electrical and computer engineering from UBC in 1985, 1988, and 1994, respectively. He was president of Daikin Comtec US (founder of DVD) and executive vice president of Sonic Solutions. He is a registered professional engineer in British Columbia and has been an active Member of the SCC, IEEE, and the Association for Computing Machinery. He is presently the director of the Institute for Computing, Information and Cognitive Systems (160 faculty members and more than 1,000 graduate students) at UBC. He is also a professor in the UBC Department of Electrical and Computer Engineering, the inaugural holder of the Dolby Pro-

fessorship in Digital Multimedia, and the current director of the Master of Software Systems Program at UBC.

REFERENCES

- [1] JCT-VC, "Encoder-side description of test model under consideration," in *Proc. JCT-VC Meeting*, Geneva, Switzerland, July 2010, JCT-VC-B204.
- [2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.
- [3] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 6," JCT-VC-H1003, Feb. 2012.
- [4] A. Fuldseth, M. Horowitz, S. Xu, A. Segall, and M. Zhou, "Tiles," JCTVC-F335, July 2011.
- [5] K. Misra, J. Zhao, and A. Segall, "Lightweight slicing for entropy coding," JCTVC-D070, Jan. 2011.
- [6] C. Gordon, F. Henry, and S. Pateux, "Wavefront parallel processing for HEVC encoding and decoding," JCTVC-F274, July 2011.
- [7] J. Chen and T. Lee, "Planar intra prediction improvement," JCTVC-F483, July 2011.
- [8] H. Li, B. Li, L. Li, J. Zhang, H. Yang, and H. Yu, "Non-CE6: Simplification of intra chroma mode coding," JCTVC-H0326, Feb. 2012.
- [9] J. Chen, V. Seregin, W.-J. Han, J. Kim, and J. Moon, "CE6.a.4: Chroma intra prediction by reconstructed luma samples," JCTVC-E266, Mar. 2011.
- [10] I.-K. Kim, W.-J. Han, J. H. Park, and X. Zheng, "CE2: Test results of asymmetric motion partition (AMP)," JCTVC-F379, July 2011.
- [11] E. Alshina, A. Alshin, J.-H. Park, J. Lou, and K. Minoo, "CE3: 7 taps interpolation filters for quarter pel position MC from Samsung and motorola mobility," JCTVC-G778, Geneva, Nov. 2011.
- [12] A. Fuldseth, G. Bjontegaard, and M. Budagavi, "CE10: Core transform design for HEVC," JCTVC-G495, Nov. 2011.
- [13] J. Han, A. Saxena, and K. Rose, "Towards jointly optimal spatial prediction and adaptive transform in video/image coding," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2010, pp. 726–729.
- [14] J. Lainema, K. Ugur, and A. Hallapuro, "Single entropy coder for HEVC with a high throughput binarization mode," JCTVC-G569, Nov. 2011.
- [15] C.-M. Fu, C.-Y. Chen, Y.-W. Huang, and S. Lei, "Sample adaptive offset for HEVC," in *Proc. Int. Workshop on Multimedia Signal Processing (MMSP)*, Oct. 2011.
- [16] P. Lai, F. C. A. Fernandes, H. Guermazi, F. Kossentini, and M. Horowitz, "CE8 Subtest 4: ALF using vertical-size 5 filters with up to 9 coefficients," JCTVC-F303, July 2011.
- [17] C.-Y. Tsai, C.-Y. Chen, C.-M. Fu, Y.-W. Huang, and S. Lei, "One-pass encoding algorithm for adaptive loop filter in high-efficiency video coding," in *Proc. Visual Communications and Image Processing (VCIP)*, Nov. 2011.
- [18] *Joint Call for Proposals on Video Compression Technology*, ISO/IEC JTCl/SC29/WG11, N11113, Jan. 2010.
- [19] F. Bossen, "Common test conditions and software reference configurations," JCTVC-F900, July 2011.
- [20] G. J. Sullivan, J.-R. Ohm, F. Bossen, and T. Wiegand, "JCT-VC AHG report: HM subjective quality investigation," JCTVC-H0022, Feb. 2012.