Intra Prediction and Mode Coding in VVC

J. Pfaff, A. Filippov, S. Liu, *Senior Member, IEEE*, X. Zhao, *Member, IEEE*, J. Chen, *Senior Member, IEEE*, S. De-Luxán-Hernández, T. Wiegand, *Fellow, IEEE*, V. Rufitskiy, A. K. Ramasubramonian and G. Van der Auwera, *Senior Member, IEEE*

***Abstract*—This paper presents the intra prediction and mode coding of the Versatile Video Coding (VVC) standard. This standard was collaboratively developed by the Joint Video Experts Team (JVET). It follows the traditional architecture of a hybrid block-based codec that was also the basis of previous standards. Almost all intra prediction features of VVC either contain substantial modifications in comparison with its predecessor H.265/HEVC or were newly added. The key aspects of these tools are the following: 65 angular intra prediction modes with block shape-adaptive directions and 4-tap interpolation filters are supported as well as the DC and Planar mode, Position Dependent Prediction Combination is applied for most of these modes, Multiple Reference Line Prediction can be used, an intra block can be further subdivided by the Intra Subpartition mode, Matrix-based Intra Prediction is supported, and the chroma prediction signal can be generated by the Cross Component Linear Model method. Finally, the intra prediction mode in VVC is coded separately for luma and chroma. Here, a Most Probable Mode list containing six modes is applied for luma. The individual compression performance of tools is reported in this paper. For the full VVC intra codec, a bitrate saving of 25% on average is reported over H.265/HEVC using an objective metric. Significant subjective benefits are illustrated with specific examples.**

***Index Terms*—Versatile Video Coding, intra prediction**

I. Introduction

VERSATILE Video Coding (VVC) [1], [2] is an international video coding standard that enables increased compression capabilities compared with Advanced Video Coding (AVC) [3], [4] and High Efficiency Video Coding (HEVC) [5], [6]. It has been collaboratively designed by the Joint Video Experts Team (JVET) that was founded by two international standardization organizations, ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG). This paper describes intra prediction and mode coding in the VVC standard.

The VVC design follows the principle of a hybrid, block-based video codec that was also employed in the previous standards H.264/AVC and H.265/HEVC. Here, a video frame is subdivided into blocks. For each block, a prediction is performed and a transform, which may be the identity mapping, is applied to the residual. This is followed by entropy coding of the prediction mode information and the transform coefficients. The prediction is generated by inter prediction or by intra prediction. While the former makes use of temporal redundancies, the latter exploits spatial correlations to generate the prediction signal for a given block out of previously reconstructed samples of the same frame that are located close to the current samples.

Intra prediction forms a cornerstone for applications of the VVC standard. It arises in the following two scenarios. First, in order to guarantee random access capability (excluding the gradual decoding refresh scenario), frames on which only intra prediction, but not inter prediction is allowed, so-called I-frames, are inserted into each video sequence. Second, local temporal scene changes, for example occlusions, may occur within a video frame. Here, inter prediction may not be effective and thus intra prediction may be applied to those regions. Although in typical video sequences, inter prediction is used most frequently, the blocks coded in intra prediction mode contribute to a significant part of the overall bitrate. Moreover, efficient intra prediction tools can mitigate bitrate fluctuations that typically occur with every I-frame due to the higher bitrate it consumes.

Compared to its predecessor HEVC [7], most of the intra prediction tools of VVC either include substantial modifications or are newly added. It is important to point out that the prediction from already reconstructed reference samples of the same frame introduces sequential dependencies between the intra predicted blocks. Consequently, in contrast to motion compensated prediction, intra predicted blocks in a frame cannot always be computed in parallel at a decoder and thus they form a particular challenge for real-time decoding. This phenomenon, often referred to as the 'intra critical loop', imposes tight limits on the implementation complexity of intra prediction modes. These constraints were carefully considered and incorporated into the design of all the VVC intra prediction tools.

This paper is structured as follows. In Section II, an overview of the main features of VVC intra coding, including newly added tools, is given. In Section III, a brief review of the partitioning schemes in VVC that result in intra coded blocks is provided. In Section IV, the intra prediction tools of VVC are described in detail whereas their signaling is outlined in Section V. In Section VI, experimental results are reported while conclusions are drawn in the final Section VII.

II. Overview of the Main New Intra Prediction Features in VVC

In this section, an overview of the main new intra prediction tools that are part of the VVC standard is given.

J. Pfaff, S. De-Luxán-Hernández and T. Wiegand are with the Fraunhofer Institute for Telecommunications, Heinrich Hertz Institute, Berlin, Germany. T. Wiegand is also with Technical University of Berlin, Berlin, Germany.

A. Filippov and V. Rufitskiy are with Huawei, Moscow, Russia.

S. Liu and X. Zhao are with Tencent Media Lab, Palo Alto, CA, USA.

J. Chen, A. K. Ramasubramonian and G. Van der Auwera are with Qualcomm Technologies Inc., San Diego, CA, USA.

Corresponding authors: J. Pfaff (e-mail: jonathan.pfaff@hhi.fraunhofer.de) and G. Van der Auwera (e-mail: geertv@qti.qualcomm.com)

1) Angular intra prediction with 65 angles and 4-tap interpolation filters: Angular intra prediction with 65 different directions is enabled. Two types of 4-tap interpolation and smoothing filters are used in the prediction generation.

2) Wide-Angle Intra Prediction (WAIP): For the newly added non-square/rectangular intra block shapes, angular intra prediction modes with wider angles of prediction are added in addition to those enabled for square blocks.

3) Position Dependent Prediction Combination (PDPC): Filtering with spatially varying weights is applied to blocks that use Planar and DC modes as well as certain angular modes.

4) Multiple Reference Line (MRL) prediction: The reconstructed samples used as reference samples in the angular and the DC prediction modes can be obtained from samples that lie two or three lines to the left and above a block.

5) Intra Subpartition (ISP) mode: Intra predicted blocks can be subdivided either horizontally or vertically into smaller blocks called subpartitions. On each of them, the prediction and transform coding operations are performed separately, but the intra mode is shared across all subpartitions.

6) Matrix-based Intra Prediction (MIP): The intra prediction samples can be generated by modes which perform a downsampling of the reference samples, a matrix vector multiplication and an upsampling of the result.

7) Cross Component Linear Model (CCLM): The chroma components of a block can be predicted from the collocated reconstructed luma samples by linear models whose parameters are derived from already reconstructed luma and chroma samples that are adjacent to the block.

8) Intra mode coding with 6 MPMs: The Planar mode, which is always a Most Probable Mode (MPM), is coded first with a separate flag. The DC and the angular modes are coded using a list of the remaining five MPMs that are derived from intra modes of neighboring blocks.

III. Partitioning Structures in VVC

The present section provides a brief outline of the partitioning methods of VVC, focusing on the aspects that are most relevant to intra coding. For a detailed description of the partitioning schemes, the reader is referred to [8].

In the VVC standard, a video frame is initially divided into coding tree units (CTUs) that cover squares of at most 128×128 luma samples. Each CTU is further partitioned into coding units (CUs) by a quadtree splitting followed by binary or ternary splittings. The resulting CUs are of rectangular shape where the width and height are integral powers of two. The partitioning of the CTUs can be carried out either jointly for luma and chroma, which is referred to as 'single tree', or independently for luma and chroma, which is referred to as 'dual tree'. The latter is supported only for I-slices.

Each CU is coded either in inter or in intra mode. For intra CUs, the intra mode is signaled once for the entire CU. Moreover, intra prediction and transform coding are performed at the transform block (TB) level. Each CU consists of a single TB, except in the cases of ISP and of implicit splitting. The former is described in section IV-F. The latter occurs when the size of any one side of the CU exceeds the maximum transform size; in this case, the CU is split further into multiple TBs.

For luma CUs, the maximum side length of a TB is 64 and the minimum side length is 4. It follows that luma TBs are $W \times H$ rectangular blocks of width W and height H, where $W, H \in \{4, 8, 16, 32, 64\}$. For chroma CUs, the maximum TB side length is 32 and chroma TBs are rectangular $W \times H$ blocks of width W and height H. Here, $W, H \in \{2, 4, 8, 16, 32\}$, but blocks of shapes $2 \times H$ and 4×2 are excluded in order to address memory architecture and throughput requirements.

IV. Intra Prediction Generation

A. Reference Sample Generation

In VVC, the intra prediction samples are generated using reference samples that are obtained from reconstructed samples of neighboring blocks. For a $W \times H$ block, the reference samples are constituted of the $2 \cdot H$ reconstructed samples to the left of the block, the top left reconstructed sample and the $2 \cdot W$ reference samples above the block. However, for MRL the reference samples are not directly adjacent to the block (see Section IV-E). In general, not all reference samples are available, i.e., located at already reconstructed sample positions, and unavailable reference samples are generated by a padding mechanism. This padding mechanism is very similar to the method used in HEVC. First, the availability determination for the reference samples is performed on the basis of 4×4 units of the neighboring reconstructed samples. Here, the set of available units always forms a contiguous interval of a subset of the reference samples. The outermost reconstructed samples of this interval are then used as values for the corresponding unavailable parts of the reference samples.

B. Angular Intra Prediction

Angular intra prediction is a directional intra prediction method that is supported in both AVC and HEVC and that is also part of VVC. In comparison to HEVC, the angular intra prediction of VVC was modified by increasing the prediction accuracy and by an adaptation to the new partitioning framework. The former was realized by enlarging the number of angular prediction directions and by more accurate interpolation filters, while the latter was achieved by introducing wide-angular intra prediction modes.

1) 65 Angular Modes and Wide-Angle Intra Prediction: In VVC, the number of directional modes available for a given block is increased to 65 directions from the 33 HEVC directions. The angular modes of VVC are depicted in Fig. 1. The directions having even indices between 2 and 66 are equivalent to the directions of the angular modes supported in HEVC.

For blocks of square shape, an equal number of angular modes is assigned to the top and left side of a block. On the other hand, for intra blocks of rectangular shape, which are not present in HEVC, but which are a central part of VVC's partitioning scheme as described in section III, more intra prediction directions are assigned to the longer side of a block. The additional modes allocated along a longer side are

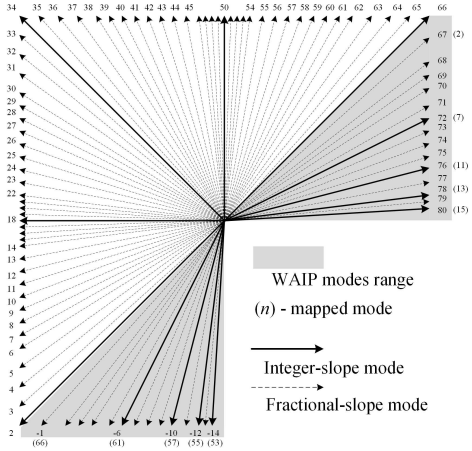


Fig. 1: Angular intra prediction modes in VVC.

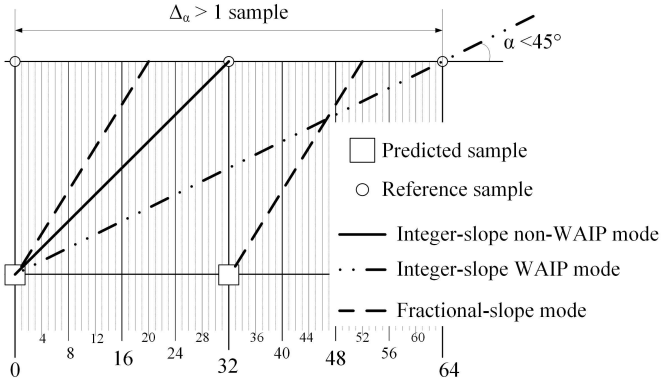


Fig. 2: Integer- and fractional-slope modes in VVC.

called Wide-Angle Intra Prediction (WAIP) modes [9], since they correspond to prediction directions with angles greater than $45°$ relative to the horizontal or vertical mode.

A WAIP mode for a given mode index is defined by mapping the original directional mode to a mode that has the opposite direction with an index offset equal to one, as shown in Fig. 1. For a given rectangular block, the aspect ratio, i.e., the ratio of width to height, is used to determine which angular modes are to be replaced by the corresponding wide-angular modes. In Table I, the last and first WAIP mode indices and, respectively, the first and last non-WAIP mode indices are enumerated for each aspect ratio that can occur for intra blocks in VVC. It can be seen from this table that the last WAIP modes, which belong to the sets $\{-14, -12, -10, -6\}$ and $\{72, 76, 78, 80\}$ correspond to the counter-diagonals of a given rectangular block and have integer slope [10]. Thus, they have similar properties as the two counter-diagonal modes 2 and 66, which determine the range of angular prediction modes in the case of square blocks in Fig 1.

For square-shaped blocks in VVC, each pair of predicted samples that are horizontally or vertically adjacent are predicted from a pair of adjacent reference samples. To the contrary, WAIP extends the angular range of directional prediction beyond $45°$, and therefore, for a coding block predicted with a WAIP mode, adjacent predicted samples may be predicted from non-adjacent reference samples [11]. To suppress discontinuities caused by this phenomenon, the reference samples are always smoothed for WAIP modes. This smoothing is performed by either enabling reference sample filtering or by selecting the smoothing interpolation filters (see section IV-B2). More detailed explanations on handling reference sample filtering for WAIP can be found in [12].

The introduction of WAIP modes does not cause any changes in the intra mode coding. Rather, for the latter, each mode in the range $[67, 80]$ is treated as its corresponding mode in the range $[2, 14]$ and each mode in the range $[-14, -1]$ is treated as its corresponding mode in the range $[53, 66]$. The mapping of a non-WAIP mode to the corresponding WAIP mode is invoked only within the process of intra prediction generation of a block.

TABLE I: Intra prediction modes corresponding to the counter-diagonals of blocks with different aspect ratios.

Aspect ratio	Counter-diagonal mode indices		Inclusive range of WAIP modes
	WAIP	non-WAIP	
16:1	80	16	67 .. 80
8:1	78	14	67 .. 78
4:1	76	12	67 .. 76
2:1	72	8	67 .. 72
1:1	-	2, 66	-
1:2	-6	60	-6 .. -1
1:4	-10	56	-10 .. -1
1:8	-12	54	-12 .. -1
1:16	-14	52	-14 .. -1

2) Reference Sample Filtering: Similar to HEVC, intra prediction in VVC has two filtering mechanisms applied to reference samples, namely reference sample smoothing and interpolation filtering. Only one of these two mechanisms can be applied on a given block in order to avoid a latency increase that would be caused by a sequential application of two filtering operations. More precisely, reference sample smoothing is applied only to integer-slope modes in luma blocks while interpolation filtering is applied to fractional-slope modes.

For reference sample smoothing, the reference samples are filtered using the finite impulse response filter $\{1, 2, 1\}/4$. Reference sample smoothing is invoked for integer-slope modes if the number of samples in the given block is more than 32.

For interpolation filtering, if a sample projection for a given prediction direction falls on a fractional position between reference samples as shown in Fig. 2, the predicted sample value is obtained by applying an interpolation filter to the reference samples around the fractional sample position. For luma blocks, 4-tap interpolation filters [13] are used and the predicted sample $\text{pred}(x, y)$ is obtained as

$$\text{pred}(x, y) = \left(\sum_{i=0}^{3} f[p][i] \cdot r[i_0 - 1 + i] + 32 \right) >> 6. \quad (1)$$

Here, i_0 denotes the closest left-side integer position of a

predicted sample projection within the reference samples r, while $p \in \{0, \ldots, 31\}$ constitutes the fractional part of the predicted sample projection with respect to i_0 at a $1/32$-pel accuracy as depicted in Fig. 2.

The interpolation filter coefficients $f[p][i]$ in (1) are signed integers whose magnitudes are stored in 6-bit precision. They either represent a DCT-based interpolation filter (DCTIF) or a 4-tap smoothing interpolation filter (SIF) and are delineated in the VVC specification. The DCTIF is constructed in the same way as the chroma DCTIF used for motion compensation in both HEVC and VVC [14]. The SIF is obtained by convolving the linear 2-tap interpolation filter of HEVC with $\{1, 2, 1\}/4$ and is thus consistent with the reference sample smoothing described above and in [15]. For the chroma components, the linear 2-tap interpolation filter of HEVC is used in VVC.

The type of the interpolation filter is not signaled in the bitstream and is determined based on the size of the block and the intra prediction mode index m. If $\min(|m-50|, |m-18|) > T$, the SIF is used, otherwise, the DCTIF is used. Here, T is a threshold that depends on the block size.

C. DC and Planar Intra Prediction Mode

As in AVC and HEVC, a DC intra prediction mode is also supported in VVC. The DC intra prediction mode of HEVC and AVC uses the mean sample value of the reference samples to the left and above the block for prediction generation. Here, since only square blocks with side lengths that are an integral power of 2 are supported, the division needed in the mean value calculation can be implemented by a bit shift.

On the other hand in VVC, although the width and the height of an intra block are integral powers of 2, this is not the case for their sum if the block is non-square. Thus, in order to avoid a division for such blocks, VVC uses the reference samples only along the longer side of a rectangular block to calculate the mean value, while for square blocks the reference samples from both sides are used. According to [16], this modification does not cause any degradation in compression performance.

The Planar intra prediction mode specified in HEVC is also enabled in VVC. In the Planar mode, the predicted sample values are obtained as a weighted average of 4 reference sample values. Here, the reference samples in the same row or column as the current sample and the reference samples on the bottom-left and on the top-right position with respect to the block are used. In VVC, the reference sample smoothing filter is also applied for the Planar mode in the luma component depending on the block size.

D. Position Dependent Prediction Combination

HEVC applies a gradient-based offset to samples of the first row or first column of the intra prediction block in case of the horizontal or vertical intra mode, which results in a reduced prediction error along the block boundary that is farthest from the prediction boundary. More generally, the initial position-dependent prediction combination (PDPC) method [17], which included optional block-level flag signaling, combines intra prediction block samples with unfiltered or filtered boundary reference samples by employing intra mode and position dependent weighting. A simplified version of PDPC [18], [19] is applied to the Planar, DC, horizontal and vertical modes without signaling. These main intra modes were determined to benefit most in terms of coding gain. The simplified PDPC, including the angular mode extension [20], was studied in the VVC core experiment on intra prediction and mode coding [21] and was adopted at the 11th JVET meeting.

The following sections describe the PDPC algorithms that are part of the VVC standard. PDPC is disabled if the block width or height is smaller than 4 samples, or if the MRL mode is used, or if BDPCM is used. Otherwise, PDPC is applied to the Planar, DC, horizontal, vertical intra modes, as well as to certain angular intra modes. The angular modes have a mode index smaller than 18 (horizontal mode) or greater than 50 (vertical mode), which includes the wide-angle modes if mode remapping is applicable based on the block shape.

If PDPC is applied, the prediction sample $P(x, y)$ located at (x, y) is calculated as follows:

$$\begin{aligned} P(x,y) \quad = \quad & (w_L R_L + w_T R_T + \\ & (64 - w_L - w_T) P(x,y) + 32) \gg 6, \end{aligned} \tag{2}$$

where w_L and w_T are the position dependent weights, R_L and R_T represent the neighboring reference samples at the left and top of the current block, respectively. The coordinate $(0, 0)$ addresses the top-left sample within the prediction block. The operations $\gg$ and $\ll$ represent binary right and left shifts, respectively. In the following, the reference samples and position dependent weights are defined for each mode.

1) PDPC applied to Planar and DC modes: For the Planar and DC modes, the reference samples R_L and R_T have the following coordinates, where $R(x, y)$ is the array of reconstructed neighboring samples:

$$R_L = R(-1, y), R_T = R(x, -1), \tag{3}$$

with $x = 0 \ldots width - 1$, $y = 0 \ldots height - 1$, and $width$ and $height$ the block dimensions.

The position dependent weights w_T and w_L are calculated as follows:

$$\begin{aligned} & w_T = 32 \gg ((y \ll 1) \gg \textit{scale}), \\ & w_L = 32 \gg ((x \ll 1) \gg \textit{scale}), \\ & \textit{scale} = (\log_2(\textit{width}) + \log_2(\textit{height}) - 2) \gg 2. \end{aligned} \tag{4}$$

2) PDPC applied to horizontal and vertical intra modes: For these modes, the neighboring reference samples are defined as follows:

$$\begin{aligned} & R_L = R(-1, y) - R(-1, -1) + P(x, y), \\ & R_T = R(x, -1) - R(-1, -1) + P(x, y), \end{aligned} \tag{5}$$

with $R(-1, -1)$ the top-left neighboring reference sample. The prediction sample $P(x, y)$ is included here in the reference sample definition to simplify Eq. 2. The position dependent weights w_T and w_L are determined as follows. For the horizontal mode, the weight w_T is determined as in Eq. 4 and $w_L = 0$, while for the vertical mode, the weight w_L is determined as in Eq. 4 and $w_T = 0$. After computing $P(x, y)$ using Eq. 2, the resulting value is clipped to the sample bit depth.

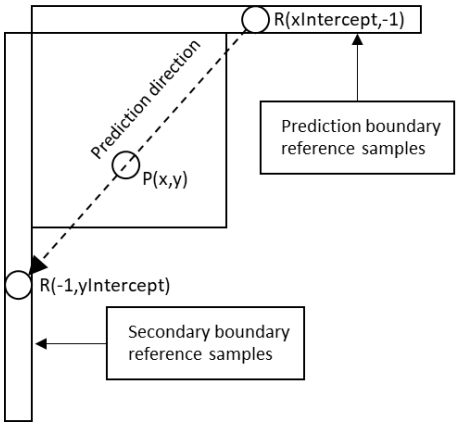


Fig. 3: Angular PDPC.

3) PDPC applied to angular intra modes: The extension of simplified PDPC to angular modes, as initially proposed in [20], computes an intercept with the neighboring reference samples (secondary boundary) that are opposite to the neighboring reference sample boundary that is used for prediction along the angular mode direction. This is illustrated in Fig. 3. An intercept only exists for positive angular mode directions and its value is limited by the length of the secondary boundary. PDPC for angular modes in [20] included the diagonal modes 2 and 66 as well as the eight adjacent modes 58 to 65 (inclusive), and modes 3 to 10 (inclusive). With the adoption of wide-angle mode remapping depending on the block shape, the mode range was extended to modes 10 and below, and modes 58 and above (excluding CCLM). At a later stage of the standardization process, PDPC for angular modes was simplified in [22] by removing a sample-based condition to check whether the intercept exceeds the size of the secondary boundary. In addition, the angular mode ranges were extended to modes 17 (adjacent to horizontal mode) and below, and to modes 51 (adjacent to vertical mode) and above.

In the VVC standard, for an angular mode in the range 51 and above (excluding CCLM), the y-intercept with the secondary reference sample boundary, as illustrated in Fig. 3, is determined as follows:

$$\textit{yIntercept} = y + (((x+1) \times \textit{invAngle} + 256) \gg 9), \quad (6)$$

$$\begin{aligned} \textit{scale} = \min(2, \log_2(\textit{height}) - \\ \text{floor}(\log_2(3 \times \textit{invAngle} - 2)) + 8), \end{aligned} \quad (7)$$

with (x, y) the coordinates of $P(x, y)$ and $invAngle$ defined as the inverse of the tangent of the prediction direction angle. The $invAngle$ values per direction are already available in VVC to compute the reference sample array mapping for negative angular modes, hence, no additional look-up table is required. If $scale$ is equal to or greater than zero, the PDPC neighboring reference samples and position dependent weights are determined as follows:

$$\begin{aligned} R_L &= \begin{cases} R(-1, \textit{yIntercept}), & \text{if } x < (3 \ll \textit{scale}) \\ 0, & \text{else} \end{cases}, \\ w_L &= 32 \gg ((x \ll 1) \gg \textit{scale}), \end{aligned} \quad (8)$$

while R_T and w_T are both equal to zero. For an angular mode in the range 17 and below, the x-intercept with the secondary reference sample boundary is computed similarly.

E. Multiple Reference Line prediction

In addition to the directly adjacent line of neighboring samples, one of the two non-adjacent reference lines that are depicted in Figure 4 can comprise the input for intra prediction in VVC. The latter use of reference samples is referred to as multiple reference line (MRL) prediction.

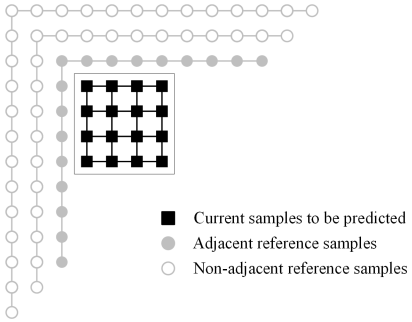


Fig. 4: Illustration of multiple reference line prediction using two non-adjacent reference lines.

1) Motivation: In HEVC and AVC, intra prediction schemes only exploit the correlation between current samples and the adjacent reference samples. However, the adjacent reference samples may not always be beneficial for generating intra prediction samples due to noise in the reference samples, object occlusion or inaccuracy of fractional interpolation. Further investigations on intra prediction schemes beyond HEVC validated the benefits of enabling non-adjacent reference lines for intra prediction [23], [24], [25], [26].

2) Algorithm Description: During the VVC standardization process, a modified design of the MRL prediction mode was proposed [27] and integrated in the VVC draft specification at the 12th JVET meeting. Its key aspects are as follows.

The intra modes that can be used for MRL are the DC mode and the angular prediction modes. However, for a given block not all of these modes can be combined with MRL. Rather, the MRL mode is always coupled with a Most Probable Mode (MPM), as described in Section V. This coupling means that if non-adjacent reference lines are used, the intra prediction mode is one of the MPMs. Such a design of an MPM-based MRL prediction mode is motivated by the observation that non-adjacent reference lines are mainly beneficial for texture patterns with sharp and strongly directed edges. In these cases, MPMs are much more frequently selected since there is typically a strong correlation between the texture patterns of the neighboring and the current blocks. On the other hand, choosing a non-MPM for intra prediction is an indication that edges are not consistently distributed in neighboring blocks, and thus the MRL prediction mode is expected to be less useful in this case. In addition, it has been observed that MRL does not provide additional coding gain when the intra prediction mode is the Planar mode, since this mode is typically used for

smooth areas. Consequently, MRL is disabled for the Planar mode, which is always one of the MPMs.

The angular or DC prediction process in MRL is very similar to the case of a directly adjacent reference line, which was described in Sections IV-B1, IV-B2 and IV-C. However for angular modes with a non-integer slope, the DCT-based interpolation filter is always used. This design choice is both evidenced by experimental results and aligned with the empirical observation that MRL is mostly beneficial for sharp and strongly directed edges where the DCTIF is more appropriate since it retains more high frequencies than the SIF.

From a hardware design perspective, applying multiple reference lines as proposed in the initial methods required extra cost of line buffers that are used for holding the additional reference lines. In typical hardware designs, line buffers are part of the on-chip memory architecture for image and video coding and it is of great importance to minimize their on-chip area. To address this issue, MRL is disabled and not signaled for the coding units that are attached to the top boundary of the CTU. In this way, the extra buffers for holding non-adjacent reference lines are bounded by 128, which is the width of the largest unit size.

3) Encoder Algorithm: The MRL design proposed in [27] also features a specific encoder implementation on top of the reference software of VVC, i.e., VTM. In the VTM, for angular intra prediction mode selection, a small set of candidate intra prediction modes is derived using a Sum of Absolute Transform Difference (SATD) metric. In a second step, a relatively more complex rate-distortion optimization is performed on this reduced set of candidate modes to select the final intra prediction mode for the current block. With MRL, only extra calculations for the first round of SATD-based evaluations of intra mode costs are added while the number of candidate modes for the second round is kept unchanged. In this way, the MRL implementation in the VTM leads to a very minor encoder runtime increase. This is a major difference to the initial designs of MRL presented above, where multiple reference lines were tested by additional full rate distortion searches which lead to a significant encoder runtime increase.

F. Intra Subpartition Mode

The Intra Subpartition (ISP) mode is one of the newly introduced tools in VVC. It is a partitioning mechanism aimed to model non-stationary characteristics of intra predicted blocks. Specifically, ISP splits the luma component of a block vertically or horizontally into K equal-sized subpartitions that are processed one-by-one in a sequential manner.

1) Layout: Given a $W \times H$ block of width W and height H, the dimensions of the corresponding subpartitions are $W \times \frac{H}{K}$ if the split is horizontal and $\frac{W}{K} \times H$ if it is vertical.

In earlier versions of the algorithm [28][29][30], K was equal to H (horizontal split) or W (vertical split), so that the resulting subpartitions would exclusively become one-dimensional blocks or lines. Although such an approach provides a benefit in terms of prediction gain [31], it does not exploit the potential 2-D transform gain [31] and it can increase the rate overhead excessively in cases where the total number of subpartitions is very large. Furthermore, lines with a length shorter than 16 samples present a hardware implementation challenge, since they require increasing the worst-case scenario throughput of HEVC due to the aforementioned 'intra critical loop'. In order to address these issues, VVC utilizes the configuration described in essence in [32], which disallows ISP for 4×4 blocks, sets $K = 2$ for 4×8 or 8×4 blocks and sets $K = 4$ in all other cases.

Figure 5 illustrates the splits for the generic case ($K = 4$). By the definition of K, it is guaranteed that all subpartitions have at least 16 samples, thereby keeping the worst-case throughput scenario for intra predicted blocks at the same level as in HEVC. On the other hand, ISP allows the generation of transform block (TB) sizes equal to $1 \times H$, $2 \times H$, $W \times 1$ and $W \times 2$, which were not present in AVC or HEVC.

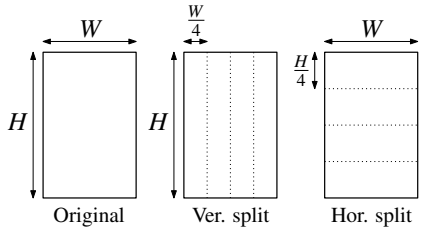


Fig. 5: ISP splits in the $K = 4$ case.

2) Processing: The subpartitions are processed one-by-one from the top to the bottom if the split is horizontal and from the left to the right if it is vertical. Despite the fact that previous versions of the algorithm [28][29][30] incorporated reversed variants of these arrangements (from the bottom to the top and from the right to the left) for certain combinations of split type and angular intra mode, they were finally not included in the standard to avoid an increase in the implementation complexity.

Each subpartition is coded in the same manner as any other intra predicted block: first, a prediction is generated using the neighboring samples of the subpartition and the corresponding residual signal is obtained. The latter is transformed and quantized and the transform coefficients are entropy coded and sent to the decoder. Finally, the resulting reconstructed samples can be employed to generate the prediction of the next subpartition. This procedure continues until all subpartitions have been coded.

However, there is an exception in the aforementioned process for subpartitions whose width is smaller than 4. This is due to the fact that typical hardware architectures allocate samples using a raster scan pattern [33] and then access them in 4×1 groups. For such designs, $1 \times H$ and $2 \times H$ subpartitions could constitute an issue. As a consequence, a minimum prediction width of 4 samples is enforced for ISP [34] as follows: instead of performing each prediction at the TB level, TBs are grouped in $4 \times H$ regions that are predicted at once using only the adjacent samples as reference. This guarantees that the TBs within each $4 \times H$ region can be processed at the same time. For instance, a vertically split 4×16 CU is predicted at once and then the corresponding four 1×16 TBs are processed in parallel.

3) Prediction: Within a CU that uses ISP, all the subpartitions apply the same intra prediction mode, which therefore only needs to be coded once. The ISP intra prediction modes can be selected from any of the conventional intra prediction modes, i.e., Planar, DC and the angular modes, where for the latter modes, the conversion of an angular mode into a wide-angle mode is performed depending on the CU aspect ratio. PDPC is applied to all the ISP subpartitions in the same way as in the non-ISP case, provided that their width and height are at least 4. In case of 4-tap interpolation filtering, the DCTIF is always used.

4) Transforms: To reduce signaling costs, the implicit version of the Multiple Transform Selection (MTS) [35] is always used for ISP. Second, the Low Frequency Non-Separable Transform (LFNST) [35] is signaled globally for the whole ISP block, and thus, the same LFNST matrix is utilized for all the subpartitions that have a non-zero Coded Block Flag (CBF) [36]. LFNST can only be used for an ISP-CU if all the subpartitions fulfill the *zero-out* condition. The LFNST transform kernels for non-ISP cases are also used for ISP. Thus, LFNST cannot be utilized for ISP if the subpartitions have a width or height smaller than 4.

5) Coefficient Coding: CBF coding is adapted for the subpartitions as follows: first, in order to exploit statistical dependencies between CBFs, the context of each CBF is the value of the CBF of the previously decoded subpartition where a default value of zero is assumed for the first one. Second, if the CBFs of the first $K-1$ subpartitions are zero, then the CBF of the last subpartition is inferred to be one and thus, the all-zero CBF case is forbidden for ISP. Finally, four new significance map subblock sizes, namely 1×16, 2×8, 16×1 and 8×2, are defined for the new TB sizes $1\times H$, $2\times H$, $W\times 1$ and $W\times 2$, respectively.

G. Matrix-based Intra Prediction

The Matrix-based Intra Prediction (MIP) modes of VVC represent a new concept of intra predictors that are designed by data-driven methods. An overview of the whole MIP process is given in Figure 6. If W and H denote the width and height of the given block, the input for MIP is comprised by the W reference samples refT directly located above the block and the H reference samples refL directly located left of the block. Out of this input, the MIP intra prediction samples are generated by applying an averaging followed by a matrix-vector multiplication and a linear interpolation.

The number $\mathrm{n_{MIP}}$ of MIP modes supported on a given transform block is equal to 16 for $\mathrm{mipSizeId}=0$, equal to 8 for $\mathrm{mipSizeId}=1$ and equal to 6 for $\mathrm{mipSizeId}=2$. Here, $\mathrm{mipSizeId}$ is set to 0 for 4×4 blocks, to 1 for 8×8 blocks and for blocks that have exactly one side of length 4, and to 2 for all other blocks. Each MIP mode can be transposed which is determined by a flag $\mathrm{mipTranspose}$.

1) Averaging: In the averaging step, the top and left boundary samples refT and refL are reduced to smaller boundaries redT and redL of size $\mathrm{boundarySize}=2$ for 4×4 blocks and of size $\mathrm{boundarySize}=4$ for all other blocks. If $W=\mathrm{boundarySize}\cdot 2^n$, then redT is defined as

$$\mathrm{redT}[i]=\Big(\sum_{j=0}^{2^n-1}\mathrm{refT}[2^n\cdot i+j]+(1<<(n-1))\Big)>>n,$$

where $0\le i<\mathrm{boundarySize}$. The left reduced boundary redL is defined analogously. The two boundaries redT and redL are concatenated to the single reduced boundary pTemp. Here, redT is taken first if the mode is not transposed, and redL is taken first, otherwise. In order to reduce its magnitude for typical signal characteristics, pTemp is transformed to a vector p of equal size by

$$\mathrm{p}[0]=2^{B-1}-\mathrm{pTemp}[0];\quad \mathrm{p}[i]=\mathrm{pTemp}[i]-\mathrm{pTemp}[0],$$

where B denotes the bit-depth and $0<i<2\cdot\mathrm{boundarySize}$.

2) Matrix Vector Multiplication: In the second step, out of the reduced boundary, a reduced prediction signal $\mathrm{predMip}$ of size $\mathrm{predSize}\cdot\mathrm{predSize}$ is generated by matrix-vector multiplication, where $\mathrm{predSize}(W,H)=4$, if $\mathrm{mipSizeId}(W,H)\in\{0,1\}$ and $\mathrm{predSize}(W,H)=8$, otherwise. For the k-th MIP prediction mode, $0\le k<\mathrm{n_{MIP}}$, one computes

$$\mathrm{predMip}=(A_k\cdot\mathrm{p}+32\cdot\mathbf{1})>>6+\mathrm{pTemp}[0]\cdot\mathbf{1}. \tag{9}$$

In this equation, A_k is a matrix whose number of rows is $\mathrm{predSize}\cdot\mathrm{predSize}$ and whose number of columns equals the size of p. Moreover, '$\cdot$' denotes matrix-vector multiplication, $\mathbf{1}$ denotes the vector of ones having size as $\mathrm{predSize}\cdot\mathrm{predSize}$ and the right shift is applied elementwise. A clipping to the range $[0,2^B-1)$ is applied to the components of $\mathrm{predMip}$ afterward.

If the horizontal and vertical upsampling factors are set to $\mathrm{upHor}={}^{W}/_{\mathrm{predSize}}$ and $\mathrm{upVer}={}^{H}/_{\mathrm{predSize}}$, the signal $\mathrm{predMip}$ defines the final prediction signal pred at every upHor-th and, respectively, upVer-th sample position of the block. More precisely, if $0\le x<\mathrm{predSize}$ and $0\le y<\mathrm{predSize}$, for $\mathrm{mipTranspose}=0$, pred is defined as:

$$\begin{aligned}&\mathrm{pred}[(x+1)\cdot\mathrm{upHor}-1,(y+1)\cdot\mathrm{upVer}-1]\\&=\mathrm{predMip}[y\cdot\mathrm{predSize}+x].\end{aligned}$$

For $\mathrm{mipTranspose}=1$, pred is defined by interchanging x and y on the right-hand side of the last equation.

3) Linear Interpolation: In the final step, at the remaining sample positions, the values of pred are derived by linear interpolation, where first a horizontal and then a vertical interpolation are performed. Here, for the horizontal interpolation, the prediction is extended to the left by the reference samples.

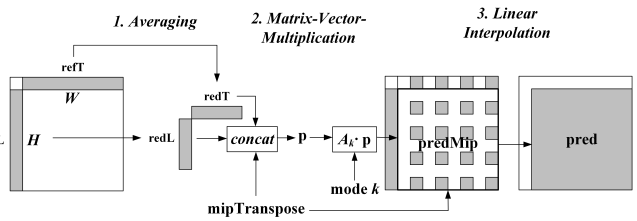


Fig. 6: Flowchart of MIP prediction.

4) Specification of the MIP Matrices: Each matrix A_k occurring in (9) is uniquely determined by $\mathrm{mipSizeId}$ and by the MIP-mode k. Its entries can be represented using 7-bit precision. Moreover, for $\mathrm{mipSizeId} = 2$, the first column of each A_k is zero since the corresponding MIP mode maps the constant boundary signal to the constant prediction signal of the same value. Thus, overall, 16 matrices of size 16×4, used for $\mathrm{mipSizeId} = 0$, 8 matrices of size 16×8, used for $\mathrm{mipSizeId} = 1$, and 6 matrices of size 64×7, used for $\mathrm{mipSizeId} = 2$, are specified for MIP.

5) Computational Complexity and Memory Requirement of MIP: The average number of multiplications per sample required to generate a MIP prediction is at most four and thus not larger than for the conventional intra prediction modes of VVC. Consequently, it can be observed that MIP, although used on more than 20% of intra blocks [37], does not yield any decoder runtime overhead, see Table III. Moreover, by the previous section, the total memory requirement to store the entries of all MIP-matrices A_k is 4144 bytes.

6) MIP as a Video-Coding Tool Designed by Data-Driven Methods: The final design of the MIP-modes in VVC can be regarded as a low-complexity variant of neural network-based intra prediction modes that were initially presented in [30], [38], [39]. While key aspects of the data-driven training algorithm of [30] were also employed to determine the parameters of the matrices A_k, the design finally adopted to VVC represents a different operational point in the gain-complexity tradeoff than the initial variant of [30]. The central steps for the complexity reduction are the downsampling of the input and the upsampling of the output of the prediction. In this way, the size of the matrices can be reduced considerably and the same matrices can be used for different block shapes. For further details on MIP and its relation to more general data-driven intra predictors, the reader is referred to [40].

H. Cross Component Linear Model

For typical video content, local dependencies can be observed between the signals of its different color components. In the HEVC range extensions, the cross component prediction (CCP) exploits these dependencies with a linear luma-to-chroma prediction in the residua domain, where the parameters of the model are signaled in the bit stream [41], [42]. In VVC, the Cross Component Linear Mode (CCLM) mode makes use of inter-channel dependencies by predicting the chroma samples from reconstructed luma samples. This prediction is carried out using a linear model in the form

$$P(i,j) = a \cdot rec'_L(i,j) + b. \tag{10}$$

Here, $P(i,j)$ represents the predicted chroma samples in a CU and $rec'_L(i,j)$ represents the reconstructed luma samples of the same CU which are downsampled for the case of non-4:4:4 color format. The model parameters a and b are derived based on reconstructed neighboring luma and chroma samples at both encoder and decoder side without explicit signaling. Such a prediction process was initially investigated in [43].

Three CCLM modes, CCLM_LT, CCLM_L and CCLM_T, are specified in VVC. These three modes differ with respect to the locations of the reference samples that are used for model parameter derivation. Samples from the top boundary are involved in the CCLM_T mode and samples from the left boundary are involved in the CCLM_L mode. In the CCLM_LT mode, samples from both the top boundary and the left boundary are used.

Overall, the prediction process of CCLM modes consists of three steps: 1) Downsampling of the luma block and its neighboring reconstructed samples to match the size of corresponding chroma block, 2) Model parameter derivation based on reconstructed neighboring samples, 3) Application of the model equation (10) to generate the chroma intra prediction samples.

1) Downsampling of the Luma Component: To match the chroma sample locations for 4:2:0 or 4:2:2: color format video sequences, two types of downsampling filter can be applied to luma samples, both of which have a 2-to-1 downsampling ratio in the horizontal and vertical directions [44]. These two filters correspond to "type-0" and "type-2" 4:2:0 chroma format content, respectively and are given by

$$f_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad f_2 = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix}. \tag{11}$$

Based on the SPS-level flag information, the 2-dimensional 6-tap or 5-tap filter is applied to the luma samples within the current block as well as its neighboring luma samples. An exception happens if the top line of the current block is a CTU boundary. In this case, the one-dimensional filter $[1,2,1]/4$ is applied to the above neighboring luma samples in order to avoid the usage of more than one luma line above the CTU boundary.

2) Model Parameter Derivation Process: The model parameters a and b from (10) are derived based on reconstructed neighboring luma and chroma samples at both encoder and decoder side to avoid any signaling overhead. In the initially adopted version of the CCLM mode, the linear minimum mean square error (LMMSE) estimator was used for derivation of the parameters [45]. In the final design, however, only four samples are involved to reduce the computational complexity [46], [47]. Fig. 7 shows the relative sample locations of an $M \times N$ chroma block, the corresponding $2M \times 2N$ luma block and their neighboring samples. In this figure, also the four samples used in the CCLM_LT mode are shown, which are marked by triangular shape. They are located at the positions of $M/4$ and $M \cdot 3/4$ at the top boundary and at the positions of $N/4$ and $N \cdot 3/4$ at the left boundary. In CCLM_T and CCLM_L modes, the top and left boundary are extended to a size of (M+N) samples, and the four samples used for the model parameter derivation are located at the positions $(M+N)/8$, $(M+N)\cdot 3/8$, $(M+N)\cdot 5/8$ and $(M+N)\cdot 7/8$.

Once the four samples are selected, four comparison operations are used to determine the two smallest and the two largest luma sample values among them. Let X_l denote the average of the two largest luma sample values and let X_s denote the average of the two smallest luma sample values. Similarly, let Y_l and Y_s denote the averages of the corresponding chroma sample values. Then, the linear model parameters are obtained

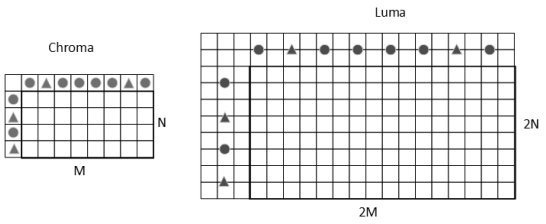


Fig. 7: Locations of the corresponding chroma and luma samples of "type-0" content.

according to the following equation:

$$\begin{aligned} a =& \frac{Y_l - Y_s}{X_l - X_s}, \\ b =& Y_s - a \cdot X_s. \end{aligned} \tag{12}$$

In this equation, the division operation to calculate the parameter a is implemented with a look-up table. To reduce the memory required for storing this table, the diff value, which is the difference between the maximum and minimum values, and the parameter a are expressed by an exponential notation. Here, the value of diff is approximated with a 4-bit significant part and an exponent. Consequently, the table for 1/diff only consists of 16 elements. This has the benefit of both reducing the complexity of the calculation and of decreasing the memory size required for storing the tables.

V. Intra Mode Coding

In HEVC, 33 angular intra prediction modes together with the DC and the Planar mode are supported and the luma intra prediction mode is signaled using 3 Most Probable Modes (MPM) [48]. On the other hand in VVC, the ISP, MRL and MIP configuration of a given luma block is signaled, and the Planar, DC and 65 angular modes are coded using an MPM scheme with 6 MPMs, due to the increased number of angular modes. Here, the Planar mode is always an MPM mode that is signaled with a separate flag [49].

1) Intra Mode Coding For Luma: The coding of the luma intra mode in VVC is summarized in Fig. 8 where the syntax elements associated with MIP, MRL, ISP and conventional intra prediction modes (Planar, DC and angular intra prediction modes) are illustrated.

As shown in Fig. 8, the flag indicating whether MIP is applied, i.e., mip_flag, is signaled first. If MIP is signaled as not being applied, the index mrl_index is signaled that indicates which reference line is to be used. If the adjacent reference line is applied, i.e., if mrl_index is 0, then the flag isp_flag indicating whether ISP is applied is signaled. When isp_flag is signaled as true, an additional syntax element isp_split_flag that indicates whether horizontal or vertical splitting is applied for ISP mode is signaled. Afterward, the syntax elements associated with conventional intra prediction modes are signaled. Here, if mrl_index is not 0, an MPM index is directly signaled since MRLP is only allowed for MPMs, as described in Section IV-E.

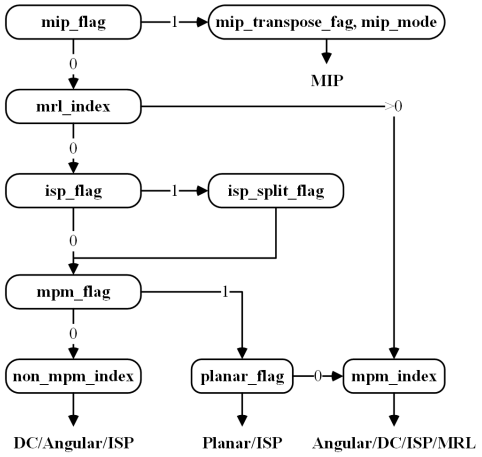


Fig. 8: Flowchart of luma intra mode coding.

For the coding of non-MIP modes, four syntax elements are defined. A first mpm_flag specifies whether an MPM is being used. If the mpm_flag is signaled as 1, a second flag Planar_flag is signaled to specify whether the MPM is the Planar mode or not. If the Planar mode is not applied, an index mpm_index is signaled to identify which of the five non-Planar MPMs, defined below, is applied. If the mpm_flag is signaled as 0, an index non_mpm_index is signaled to indicate which of the remaining 61 non-MPM modes is applied.

The isp_split_flag is entropy coded using one context. For the entropy coding of the mpm_flag one context is applied, while two contexts are applied for the entropy coding of the Planar_flag, depending on whether ISP is used or not. For the entropy coding of mpm_index, a truncated unary code is used where all bins are coded in bypass mode, i.e., no context is applied. For the entropy coding of non_mpm_index, truncated binary coding is used where all bins are coded in bypass mode as well.

For the coding of MIP modes, two separate syntax elements are signaled. First, a flag mip_transpose_flag is signaled that determines whether the transposed MIP mode is to be used or not. Second, an index mip_mode is signaled that specifies which MIP mode is to be applied. The entropy coding of mip_flag uses a context that is derived according to the mip_flag values of top and left neighboring blocks and according to the aspect ratio of the given block. The mip_transpose_flag flag is coded in bypass mode, and the index mip_mode is signaled using a truncated binary code where all bins are coded in bypass mode.

2) Construction of MPMs: The list of MPMs is constructed using default intra modes, intra modes of neighboring blocks and intra modes derived from the latter modes [50], [51]. From the set of neighboring blocks, only two blocks, namely the block next to the top-right and the block next to the bottom-left position of the current block are used in the MPM list construction. The intra prediction modes of these neighboring blocks shall be denoted as A and L. Moreover, the maximum and minimum values of A and L shall be denoted as Max and Min. Finally, the Horizontal and Vertical intra prediction modes shall be denoted as H and V. When a neighboring

block is not an intra block or a MIP-block, the Planar mode is assigned as the intra prediction mode for this neighboring block.

For the MPM list construction process, one differentiates between the following four cases:

1) Neither A nor L is an angular intra prediction mode.
2) Either A or L is an angular intra prediction mode, but not both of them are an angular intra prediction mode.
3) Both A and L are angular intra prediction modes but A is not equal to L.
4) Both A and L are angular modes and they are equal.

In the first case, the MPMs are the set {Planar, DC, V, H, V-4 and V+4} of default intra modes. In the second case, the MPMs are derived as {Planar, Max, Max-1, Max+1, Max-2 and Max+2}, and in the fourth case, the MPMs are derived as {Planar, L, L-1, L+1, L-2 and L+2}. Finally, in the third case, the first three MPMs are {Planar, L, A}, and the remaining three MPMs are derived as follows, depending on the distance between Min and Max. First, when Max-Min is equal to 1, the three remaining MPMs are {Min-1, Max+1, Min-2}. Second, when Max-Min is greater than or equal to 62, the three remaining MPMs are {Min+1, Max-1, Min+2}. Third, when Max-Min is equal to 2, the three remaining MPMs are {Min+1, Min-1, Max+1}. In all other cases, the three remaining MPMs are {Min-1, Min+1, Max-1}.

In the above formulas, a modulo operation is performed to ensure that a derived intra prediction mode is within the range of available intra prediction mode indices. For example, the derived mode L-1 is computed as 2 + ((L + 61) % 64) and L+1 is computed as 2 + ((L - 1) % 64).

3) Intra Mode Coding For Chroma: The intra prediction modes enabled for the chroma components in VVC are the three CCLM modes, CCLM_LT, CCLM_L and CCLM_T, the Planar, DC and angular intra prediction modes and MIP under a certain condition [52].

In the signaling of the chroma intra mode, a flag indicating whether CCLM is applied is signaled first. If the flag is signaled as true, it is signaled which of the three CCLM modes is applied.

In the non-CCLM case, an index intraChromaPredModeIdx ranging from 0 to 4 is signaled. When this index is equal to 4, it represents the direct mode (DM) and is binarized as one bit. Otherwise, it is binarized by three bits using a fixed length codeword that determines which of the four applicable non-DM modes is to be used. When the DM mode is used, the intra prediction mode of the collocated luma block determines the chroma intra mode as follows. If the collocated luma block uses the Planar, DC or an angular mode, the same mode is used. If the collocated luma block is coded using Intra Block Copy (IBC) or Palette mode, the DC mode is used. If the collocated luma block is coded using Block DPCM (BDPCM) mode, depending on the direction of the BDPCM, either the Horizontal or the Vertical intra prediction mode is used. If the collocated luma block uses MIP, then, if the chroma color format is 4:4:4 and the single partitioning tree is applied, the same MIP mode is applied for the chroma block [52] and otherwise, the Planar mode is applied.

When the DM is not used, the other four non-DM modes are either given by the list $\{0, 50, 18, 1\}$ or, in cases where the DM mode already belongs to that list, by modifications of it [1]. Finally, when the adaptive color transform (ACT) is applied for a block, it was observed that the intra prediction modes among different color components are highly correlated. Consequently, the intra prediction mode for chroma is not signaled and implicitly set to the DM in this case.

VI. Experimental Results

The performance evaluation of intra prediction and mode coding in VVC with respect to HEVC has been conducted using their respective reference softwares. The VVC reference software 9.0 [53] (referred to as VTM-9.0) was used to encode the sequences to generate VVC bitstreams using the Main 10 profile. A description of algorithms used in VTM-9.0 (including encoding algorithms) is available in the VVC test model document [54]. The HEVC bitstreams are obtained by configuring the HM reference software 16.0 [55] (referred to as HM-16.0) to code the sequences using the Main 10 profile. The "All Intra" (AI) coding configuration [56], where each picture is coded as one intra slice, is used for the evaluation. Several sequences with different characteristics are used for the evaluation, starting from UHD sequences to WQVGA sequences, and also content that resembles screen content (Class F). The sequences are represented in the 8-bit or 10-bit 4:2:0 format, but are encoded as 10-bit sequences. The metric that is used for evaluation of the algorithms is the PSNR-based Bjøntegaard delta rate (ΔBDR) [57], [58]. A negative (positive) ΔBDR percentage value indicates an increase (decrease) in the compression efficiency performance. The VVC and HEVC bitstreams are generated using the quantization parameter values 22, 27, 32 and 37.

A. Comparison With HEVC

The compression performance of VVC with respect to HEVC is provided in Table II. The luma and chroma BD-rates (ΔBDR) for the tested sequences are provided as an average for each class of sequences and the overall average. As can be observed, VVC provides nearly 25% BD-rate improvement over HEVC for the AI configuration, with sequences such as ParkRunning, Campfire and BasketballDrill providing over 30% luma ΔBDR. VVC has larger gain for high-resolution sequences such as UHD sequences (Classes A1 and A2) when compared with HEVC. Overall, when compared with HEVC, there is a clearly significant benefit in the coding efficiency for VVC over all classes and types of sequences. The readers may refer to [2] for a more detailed comparison of the compression efficiency of VVC and HEVC.

B. Tool-Off Tests

The compression performance of various tools in the VTM are measured using the "tool-off" test procedure, similar to the procedure followed by an ad-hoc group established by the JVET (AhG13), which tracked the performance of the various tools over the course of the development of the standard. The

TABLE II: Comparison of VVC and HEVC for the "All Intra" configuration.

Class	Resolution	ΔBDR		
		Y	U	V
Class A1	3840×2160	-29.1%	-32.4%	-34.4%
Class A2	3840×2160	-29.3%	-24.0%	-21.1%
Class B	1920×1080	-21.7%	-26.9%	-30.8%
Class C	832×480	-22.5%	-18.9%	-22.7%
Class E	1280×720	-25.8%	-26.1%	-24.5%
Class D	416×240	-18.4%	-13.3%	-13.2%
Class F	1024×768	-39.4%	-39.9%	-42.5%
A1,A2,B,C,E (Avg.)		-25.1%	-25.4%	-26.9%
All Classes (Avg.)		-26.3%	-25.8%	-27.2%

"tool-off" test for a particular tool is conducted by comparing the VTM software configured to disable the particular tool with the regular VTM software (where the tool would be enabled). All the other tools are normally enabled in both cases, thus giving an estimate of the coding efficiency improvement provided by a particular tool to the VTM software. The performance of a tool is measured using the ΔBDR for the three components (Y, U and V). Since coding tools provide compression gains, disabling a tool typically results in a loss which is indicated by a positive ΔBDR (expressed as %) for these "tool-off" tests. The performance of a tool is also dependent on the encoder algorithms used in the VTM [54]. The performance is obtained by averaging the performance of sequences in classes A1, A2, B, C and E.

Seven "tool-off" tests - Tests 1 to 7 - are performed and the results are tabulated in Table III. The tools specified in Tests 1-5 are described in earlier sections; Test 6 provides the performance when the I-slices are coded with single tree instead of dual tree partitioning; Test 7 uses the single tree configuration and all the tools in Tests 1 to 5 are switched off. Tests 1-5 are tested with the dual-tree structure for luma and chroma enabled for I-slices. Explicit tool-off tests for other intra prediction/coding tools are not presented here due to the difficulty in specifying a fair test condition in those cases.

The performance of these tests are provided in the Table III, including encoding/decoding runtime comparison. The CCLM tool (Test 1) provides considerable improvement in the coding efficiency by taking advantage of the correlation of the luma and chroma sample values - around 1.5% ΔBDR gain on luma and around 14% ΔBDR gain on the chroma components. The linear prediction model of the chroma samples from the reconstructed luma samples provides a good prediction, resulting in much smaller residual values to be encoded and hence the relatively large coding gain for the chroma components. The MIP tool (Test 2) provides gains for all classes of sequences, particularly for certain larger resolutions. The PDPC tool (Test 3) provides consistently better performance for all classes of sequences, with over 1% ΔBDR improvement for several sequences. The ISP tool (Test 4) provides significant gains for smaller resolutions (0.7% on an average, with maximum of 1.0% for one of the sequences tested). The average ΔBDR gain obtained by ISP for the A1, A2, B, C and E classes is increased with respect to Test 4 in higher QP ranges (in particular, it is approximately doubled for the 32-47 range), as shown in [59]. The MRL tool (Test 5) also provides better compression by considering additional reference samples lines, particularly for the sequences BasketballDrill and BQTerrace. Finally, dual tree partitioning provides a gain of 0.4% for luma and over 9% for chroma components. The combined gain provided by all the six tools (Test 7) is around 4% for luma and over 25% for chroma. For the Random Access configuration [56], the combined gain for the all six tools (Test 7) is around 2.4% for luma and over 15% for chroma.

When one or more tools are disabled, the tool-off test for other tools may provide a different value for the observed loss. For example, MIP and PDPC are two tools that are not enabled for the same blocks (i.e., if a block is coded with MIP, PDPC is not applied to that block and vice versa). When MIP is turned off, the tool-off test for PDPC shows an increase in loss from 0.77% to 1.20%; similarly when PDPC is turned off, the tool-off test for MIP increases from 0.63% to 1.06%. This shows that the tool-off tests in Table III indicate the performance provided by the respective tool despite all other tools that are enabled. A VVC encoder may choose to enable one or more tools depending on the trade-off between the coding efficiency and coding complexity.

C. Subjective Comparison

An example of the subjective improvement achieved by VVC compared with HEVC is provided in this section; this example illustrates that the superior objective performance of VVC also translates to visual quality improvement when compared with HEVC. The software versions for VVC and HEVC used in the previous section are also used here; however, instead of comparing sequences coded with the same QP, bitstreams that are encoded at similar bitrates are considered. Figure 9 compares the performance of VVC with respect to HEVC for a region in the 80^{th} frame of the BasketballDrill sequence. It is to be noted that the VVC bitstream is coded at roughly 3% lower bitrate than HEVC. Despite the lower bitrate, the reconstruction of the VVC frame is considerably better compared with HEVC. More textural features are visible for VVC, and the green-painted bands have much fewer artifacts at the edges. It must be noted that contributions to the visual quality improvement come from a combination of the intra coding tools described in this paper and other tools including for example partitioning, loop filters, deblocking and entropy coding, that were jointly designed for VVC [2].

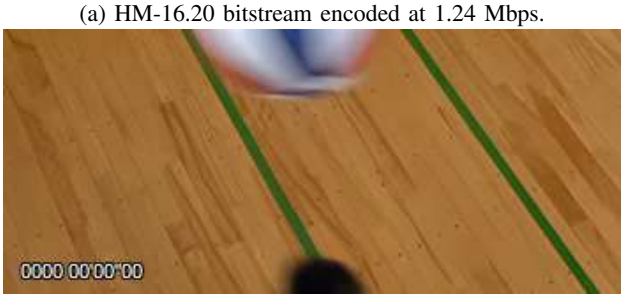
VII. Conclusions

The intra prediction tools of VVC provide a substantial compression benefit over HEVC both in an objective and in a subjective evaluation framework. Since a variety of intra prediction modes are possible for each block, compression gains are obtained throughout a large set of different types of content. On the other hand, each individual intra prediction

TABLE III: Performance of Tests 1-7

Test	Tool-off	All Intra					Random Access				
		ΔBDR			EncT	DecT	ΔBDR			EncT	DecT
		Y	U	V			Y	U	V		
Test 1	CCLM	1.54%	13.92%	14.77%	99%	96%	1.02%	11.52%	12.58%	99%	100%
Test 2	MIP	0.63%	0.20%	0.20%	89%	100%	0.33%	0.35%	0.37%	95%	100%
Test 3	PDPC	0.77%	0.50%	0.48%	98%	99%	0.39%	-0.02%	0.03%	99%	99%
Test 4	ISP	0.48%	0.30%	0.23%	85%	98%	0.28%	0.29%	0.33%	96%	100%
Test 5	MRL	0.33%	0.15%	0.14%	99%	100%	0.17%	0.08%	0.10%	100%	100%
Test 6	Dual-tree	0.41%	9.86%	9.52%	156%	103%	0.12%	4.12%	4.57%	102%	100%
Test 7	All above	4.07%	25.7%	27.43%	99%	94%	2.39%	15.69%	17.57%	90%	99%

(a) HM-16.20 bitstream encoded at 1.24 Mbps.



(b) VTM-9.0 bitstream encoded at 1.2 Mbps.

Fig. 9: Subjective quality comparison of VVC and HEVC for the BasketballDrill sequence. Several texture features are visible only in the VVC coded sequence.

method supported in VVC is of moderate computational complexity. Thus, the sequential dependencies that exist between intra-predicted blocks can be handled in real-time decoding applications.

ACKNOWLEDGEMENT

The authors would like to thank all experts of the involved standardization organizations for their contribution to the design of intra prediction and mode coding in VVC.

REFERENCES

[1] ITU-T and ISO/IEC, "Versatile Video Coding," ITU-T Rec. H.266 and ISO/IEC 23090-3, 2020.

[2] B. Bross, Y. K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the Versatile Video Coding Standard and its Applications," *IEEE Transactions on Circuits and Systems for Video Technology, this issue*.

[3] ITU-T and ISO/IEC, "Advanced Video Coding for Generic Audiovisual Services," ITU-T Rec. H.264 and ISO/IEC 14496-10, vers. 1, 2003.

[4] T. Wiegand and G. J. Sullivan and G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 7 2003.

[5] ITU-T and ISO/IEC, "High Efficiency Video Coding," ITU-T Rec. H.265 and ISO/IEC 23008-2, vers. 1, 2013.

[6] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[7] J. Lainema, F. Bossen, W. Han, J. Min, and K. Ugur, "Intra Coding of the HEVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1792–1801, 2012.

[8] Y.-W. Huang, J. An, H. Huang, X. Li, S. Hsiang, K. Zhang, H. Gao, J. Ma, and O. Chubach, "Block Partitioning Structure in the VVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology, this issue*.

[9] L. Zhao, X. Zhao, S. Liu, X. Li, J. Lainema, G. Rath, F. Urban, and F. Racapé, "Wide Angular Intra Prediction for Versatile Video Coding," in *2019 Data Compression Conference (DCC)*.

[10] L. Zhao, X. Zhao, X. Li, and S. Liu, "CE3-related: Unification of angular intra prediction for square and non-square blocks," Document JVET-L0279 of JVET, Oct. 2018.

[11] F. Racape, G. Rath, F. Urban, L. Zhao, S. Liu, X. Zhao, X. Li, A. Filippov, V. Rufitskiy, and J. Chen, "CE3-related: Wide-angle intra prediction for non-square blocks," Document JVET-K0500 of JVET, July 2018.

[12] A. Filippov and V. Rufitskiy, "Non-CE3: Cleanup of interpolation filtering for intra prediction," Document JVET-P0599 of JVET, October 2019.

[13] A. Filippov, V. Rufitskiy, J. Chen, G. V. der Auwera, A. K. Ramasubramonian, V. Seregin, T. Hsieh, and M. Karczewicz, "CE3: A combination of tests 3.1.2 and 3.1.4 for intra reference sample interpolation filter," Document JVET-L0628 of JVET, October 2018.

[14] K. Ugur, A. Alshin, E. Alshina, F. Bossen, W. Han, J. Park, and J. Lainema, "Interpolation filter design in HEVC and its coding efficiency - complexity analysis," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 1704–1708.

[15] A. Filippov, V. Rufitskiy, J. Chen, and E. Alshina, "Intra Prediction in the Emerging VVC Video Coding," *Data Compression Conference (DCC)*, 2020.

[16] A. Filippov, V. Rufitskiy, and J. Chen, "CE3-related: Alternative techniques for DC mode without division," Document JVET-K0122 of JVET, July 2018.

[17] A. Said, X. Zhao, M. Karczewicz, J. Chen, and F. Zou, "Position dependent prediction combination for intra-frame video coding," in *Proc. IEEE Int. Conf. Image Process.*, Phoenix, AZ, USA, Sep. 2016, pp. 534–538.

[18] X. Zhao, V. Seregin, A. Said, K. Zhang, H. E. Egilmez, and M. Karczewicz, "Low-complexity intra prediction refinements for video coding," in *Proc. 32rd Picture Coding Symp.*, San Francisco, CA, USA, Jun. 2018, pp. 139–143.

[19] Y.-W. Chen *et al.*, "Description of SDR, HDR, and 360° video coding technology proposal by Qualcomm and Technicolor – low and high complexity versions," Document JVET-J0021 of JVET, Apr. 2018.

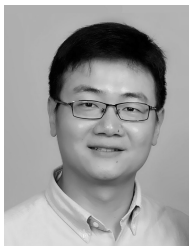
[20] G. Van der Auwera, V. Seregin, A. Said, and M. Karczewicz, "Extension

of Simplified PDPC to Diagonal Intra Modes," Document JVET-J0069 of JVET, Apr. 2018.
[21] G. Van der Auwera and J. Heo and A. Filippov, "CE3: Summary Report on Intra Prediction and Mode Coding," Document JVET-J0023 of JVET, Apr. 2018.
[22] F. Bossen, "On general intra sample prediction," Document JVET-O0364 of JVET, Jul. 2019.
[23] Y. Chang, C. Lin, P. Lin, C. Lin, and J. Tu, "Improved intra prediction method based on arbitrary reference tier coding schemes," in *2016 Picture Coding Symposium (PCS)*, 2016, pp. 1–5.
[24] J. Li, B. Li, J. Xu, and R. Xiong, "Efficient Multiple-Line-Based Intra Prediction for HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 4, pp. 947–957, 2018.
[25] Y. Chang, P. Lai, C. Lin, J. Tu, and C. Lin, "Arbitrary reference tier for intra directional modes," Document JVET-C0043 of JVET, May 2016.
[26] J. Li, B. Li, J. Xu, R. Xiong, and G. J. Sullivan, "Multiple line-based intra prediction," Document JVET-C0071 of JVET, May 2016.
[27] B. Bross *et al.*, "CE3: Multiple reference line intra prediction (Test 1.1.1, 1.1.2, 1.1.3 and 1.1.4)," Document JVET-L0283 of JVET, Oct. 2018.
[28] S. De-Luxán-Hernández, H. Schwarz, D. Marpe, and T. Wiegand, "Line-Based Intra Prediction for Next-Generation Video Coding," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 221–225.
[29] ——, "Fast Line-Based Intra Prediction for Video Coding," in *2018 IEEE International Symposium on Multimedia (ISM)*, Dec. 2018, pp. 135–138.
[30] J. Pfaff *et al.*, "Video Compression Using Generalized Binary Partitioning, Trellis Coded Quantization, Perceptually Optimized Encoding, and Advanced Prediction and Transform Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1281–1295, May 2020.
[31] J.-R. Ohm, *Multimedia Signal Coding and Transmission*, 1st ed. Springer Publishing Company, Incorporated, 2016.
[32] S. De-Luxán-Hernández, V. George, J. Ma, T. Nguyen, H. Schwarz, D. Marpe, and T. Wiegand, "An Intra Subpartition Coding Mode for VVC," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sept. 2019, pp. 1203–1207.
[33] M. Tikekar, C. Huang, C. Juvekar, V. Sze, and A. Chandrakasan, *High Efficiency Video Coding (HEVC): Algorithms and Architectures*. Springer Publishing Company, Incorporated, 2014, ch. 10: Decoder Hardware Architecture for HEVC.
[34] A. K. Ramasubramonian *et al.*, "CE3-1.6: On $1 \times N$ and $2 \times N$ subblocks of ISP," Document JVET-O0106 of JVET, Jul. 2019.
[35] X. Zhao, S.-H. Kim, Y. Zhao, H. E. Egilmez, M. Koo, S. Liu, J. Lainema, and M. Karczewicz, "Transform Coding in the VVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology, this issue*.
[36] H. Schwarz, M. Coban, M. Karczewicz, T. Chuang, F. Bossen, A. Alshin, J. Lainema, C. R. Helmrich, and T. Wiegand, "Quantization and Entropy Coding in the Versatile Video Coding (VVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology, this issue*.
[37] W. J. Chien *et al.*, "JVET AHG report: Tool reporting procedure (AHG13)," Document JVET-S0013 of JVET, Jul. 2020.
[38] J. Pfaff, P. Helle, D. Maniry, S. Kaltenstadler, W. Samek, H. Schwarz, D. Marpe, and T. Wiegand, "Neural network based intra prediction for video coding," in *Proc. SPIE 10752, Applic. of Digital Image Process. XLI*.
[39] P. Helle, J. Pfaff, M. Schäfer, R. Rischke, H. Schwarz, D. Marpe, and T. Wiegand, "Intra Picture Prediction for Video Coding with Neural Networks," in *Proc. IEEE Data Compression Conf. (DCC), Snowbird*, Mar. 2019.
[40] J. Pfaff, P. Helle, P. Merkle, M. Schäfer, B. Stallenberger, T. Hinz, H. Schwarz, D. Marpe, and T. Wiegand, "Data-driven intra-prediction modes in the development of the versatile video coding standard," *ITU Journal: ICT Discoveries*, vol. 3, no. 1, May. 2020.
[41] A. Khairat, T. Nguyen, M. Siekmann, D. Marpe, and T. Wiegand, "Adaptive cross-component prediction for 4:4:4 high efficiency video coding," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 3734–3738.
[42] D. Flynn, D. Marpe, M. Naccari, T. Nguyen, C. Rosewarne, K. Sharman, J. Sole, and J. Xu, "Overview of the Range Extensions for the HEVC Standard: Tools, Profiles, and Performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 4–19, 2016.
[43] X. Zhang, C. Gisquet, E. François, F. Zou, and O. C. Au, "Chroma Intra Prediction Based on Inter-Channel Correlation for HEVC," *IEEE Transactions on Image Processing*, vol. 23, no. 1, pp. 274–286, 2014.
[44] P. Hanhart and Y. He, "CE3: Modified CCLM downsampling filter for "type-2" content (Test 2.4)," Document JVET-M0142 of JVET, Jan. 2019.
[45] J. Chen, V. Seregin, W.-J. Han, J. Kim, and J. Moon, "CE6.a.4: Chroma intra prediction by reconstructed luma samples," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC 1/SC29/WG11, Geneva, CH, Input document JCTVC-E0266, Mar. 2011.
[46] J. Huo and others, "CE3-1.5: CCLM derived from four neighbouring samples," Document JVET-N027 of JVET, Apr. 2019.
[47] J. Li, M. Wang, L. Zhang, K. Zhang, S. Wang, S. Wang, S. Ma, and W. Gao, "Sub-Sampled Cross-Component Prediction for Chroma Component Coding," in *2020 Data Compression Conference (DCC)*, 2020, pp. 203–212.
[48] X. Zhang, S. Liu, and S. Lei, "Intra mode coding in HEVC standard," *2012 Visual Communications and Image Processing*, pp. 1–6, 2012.
[49] L. Zhao, X. Zhao, X. Li, and S. Liu, "CE3-related: Modifications on MPM list generation," Document JVET-L0279 of JVET, Jan. 2019.
[50] B. Wang, A. Kotra, S. Esenlik, H. Gao, and J. Chen, "CE3-related: A unified MPM list for intra mode coding," Document JVET-N0185 of JVET, Apr. 2019.
[51] J. Pfaff, T. Hinz, P. Merkle, P. Helle, B. Stallenberger, M. Schäfer, H. Schwarz, D. Marpe, T. Wiegand, B. Wang, S. Esenlik, A. Kotra, H. Gao, , and E. Alshina, "CE3-related: Improved intra mode coding based on unified MPM list generation," Document JVET-O0484 of JVET, June 2019.
[52] J. Pfaff, B. Stallenberger, P. Merkle, M. Schäfer, P. Helle, T. Hinz, H. Schwarz, D. Marpe, and T. Wiegand, "MIP for all channels in the case of 4:4:4-chroma format and of single tree," Document JVET-R0350 of JVET, Apr. 2020.
[53] [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tags/VTM-9.0
[54] J. Chen, Y. Ye, and S. Kim, "Algorithm description for Versatile Video Coding and Test Model 9 (VTM 9)," Document JVET-R2002 of JVET, Apr. 2020.
[55] [Online]. Available: https://vcgit.hhi.fraunhofer.de/jct-vc/HM/-/tags/HM-16.20
[56] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, "JVET common test conditions and software reference configurations for SDR video," Document JVET-N1010 of JVET, Mar. 2019.
[57] G. Bjøntegaard, "Improvements of the BD-PSNR model," ITU-T SG16 Q.6 document VCEG-AI11, Jul. 2008.
[58] K. Andersson, F. Bossen, J.-R. Ohm, A. Segall, R. Sjőberg, J. Strőm, G. J. Sullivan, and A. Tourapis, "Working practices using objective metrics for evaluation of video coding efficiency experiments (Draft 3)," Document JVET-N1010 of JVET, Jun.-Jul. 2020.
[59] S. De-Luxán-Hernández, V. George, G. Venugopal, J. Pfaff, H. Schwarz, D. Marpe, and T. Wiegand, "'Design of the Intra Subpartition mode in VVC and its optimized encoder search in VTM," in *Applications of Digital Image Processing XLI, International Society for Optics and Photonics. SPIE, 2020*, Oct. 2020.

Jonathan Pfaff received his Diploma and his Dr. rer. nat. degree in Mathematics from Bonn University, Bonn, Germany, in 2010 and 2012. After a postdoctoral research stay at Stanford University, he joined the Video Coding and Analytics Department at the Heinrich Hertz Institute in Berlin, Germany in 2015, where, since 2020, he is heading the research group on video coding technologies. He has successfully contributed to the efforts of the ITU-T Video Coding Experts Group in developing the Versatile Video Coding standard since 2017. His current research interests include image and video coding and machine learning.

Alexey Filippov received his B.Sc., Engineer, M.Sc. degrees in Electrical Engineering and Ph.D. degree in Computer Science (all with highest honor) at Vladimir State University (Vladimir, Russia) in 2001, 2002, 2003 and 2005, respectively. In parallel with pursuing his academic degrees, he made internships and worked at different organizations including Fraunhofer IIS (Erlangen, Germany). Since 2005, he served as an associate professor and the Director of the R&D center "Digital Technologies" at Vladimir State University as well as Senior Principal Architect and the Director of the Multimedia Systems Dept. at Sigma-IS (Moscow, Russia). In 2010, he received the title of associate professor for his academic contributions. Since 2012, he joined Huawei Technologies as a Principal Video Coding Engineer at the Media Algorithms laboratory of the Moscow Research Center. His scientific interests include still picture and video coding and transmission, multimedia signal processing as well as dynamically reconfigurable systems design. Since 2010, he actively contributed to the work of JPEG, MPEG, VCEG, and IETF. At the IETF Internet Video Codec working group, he led the work on the requirements used by the Alliance for Open Media (AOM) for its AV1 codec development activity. Since 2015, he was involved in the Joint Video Experts Team (JVET) activity aimed at the development of the VVC standard. In particular, he served as a coordinator of Core Experiment on Intra prediction and mode coding.

Shan Liu (M'01-SM'11) is a Tencent Distinguished Scientist and General Manager of Tencent Media Lab. She was formerly Director of Media Technology Division at MediaTek USA. She was also formerly with MERL, Sony and IBM. She has been actively contributing to international standards since the last decade and served co-Editor of HEVC SCC and the emerging VVC. She has numerous technical contributions adopted into various standards, such as HEVC, VVC, OMAF, DASH and PCC, etc. At the same time, technologies and products developed by her and under her leadership have served several hundred million users. Dr. Liu holds more than 150 granted US and global patents and has published more than 80 journal and conference papers. She received the B.Eng. degree in electronic engineering from Tsinghua University, the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, respectively. Her research interests include audio-visual, high volume, immersive and emerging media compression, intelligence, transport, and systems.



Xin Zhao (M'18) received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, and the Ph.D. degree in computer applications from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. In 2017, he joined Tencent, Palo Alto, CA, USA, where he is currently a Principal Researcher and Manager of Multimedia Standards in Tencent Media Lab. Before 2017, he was a Staff Engineer with Qualcomm, San Diego, CA, USA. He has been actively contributing to the development of multiple video standards for over 10 years, such as HEVC, 3D extensions to H.264/AVC and HEVC and VVC. He has contributed over 200 standard proposals and published over 30 journal and conference papers. His research interests include image and video coding, video processing, and transmission.

Jianle Chen received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2001 and 2006, respectively. He was formerly with Samsung Electronics, Qualcomm, and Huawei (USA) focusing on the research of video technologies. Since 2006, he has been actively involved in the development of various video coding standards, including the HEVC standard, its scalable, format range and screen content coding extensions, and most recently, the VVC standard under development in the Joint Video Experts Team (JVET). He has also been a main developer of the recursive partitioning structure with large block size, which is one of the key features of HEVC standard and its potential successors. He is currently a Director of the Multimedia R&D Group, Qualcomm, Inc., San Diego, CA, USA. His research interests include video coding and transmission, point cloud coding, AR/VR, and neural network compression. He was an Editor of the HEVC specification version 2 (the scalable HEVC (SHVC) text specification) and SHVC Test Model. For VVC, he has been the Lead Editor of the Joint Exploration Test Model (JEM) and VVC Test Model (VTM). He is an Editor of the VVC Text Specification.

Santiago De-Luxán-Hernández received his Dipl.-Ing. degree in Telecommunications Engineering from the Universidad de Las Palmas de Gran Canaria (Spain) in late 2012. He is currently with the Image and Video coding group, Video Coding and Analytics department at Fraunhofer Institute for Telecommunications-Heinrich Hertz Institute. He has successfully contributed to the efforts of the ITU-T Video Coding Experts Group in developing the Versatile Video Coding standard since 2017.

Thomas Wiegand (Fellow, IEEE) is a professor in the department of Electrical Engineering and Computer Science at the Technical University of Berlin and is jointly heading the Fraunhofer Heinrich Hertz Institute, Berlin, Germany. He received the Dipl.-Ing. degree in Electrical Engineering from the Technical University of Hamburg-Harburg, Germany, in 1995 and the Dr.-Ing. degree from the University of Erlangen-Nuremberg, Germany, in 2000.

As a student, he was a Visiting Researcher at Kobe University, Japan, the University of California at Santa Barbara and Stanford University, USA, where he also returned as a visiting professor. He was a consultant/co-founder with a number of startup companies.

Since 1995, he has been an active participant in standardization for multimedia with many successful submissions to ITU-T and ISO/IEC. In 2000, he was appointed as the Associated Rapporteur of ITU-T VCEG and from 2005-2009, he was Co-Chair of ISO/IEC MPEG Video. Since 2018, he has been appointed the chair of the ITU/WHO Focus Group on Artificial Intelligence for Health.

The projects that he co-chaired for the development of the H.264/MPEG-4 AVC standard have been recognized by the ATAS Primetime Emmy Engineering Award and a pair of NATAS Technology and Engineering Emmy Awards. For his research, he received numerous awards and multiple best paper awards for his publications. Thomson Reuters named him in their yearly list of The World's Most Influential Scientific Minds as one of the most cited researchers in his field. He has been elected to the German National Academy of Engineering (Acatech) and the National Academy of Science (Leopoldina).

Vasily Rufitskiy received his B.Sc., Engineer and M.Sc. degrees in Electrical Engineering at Vladimir State University in 2005, 2006 and 2007, respectively. Since 2005, he worked for the Multimedia Systems Dept. at Sigma-IS (Moscow, Russia). In 2012 he joined Huawei Technologies as a Senior Video Coding Engineer at Media Algorithms laboratory of Moscow Research Center. Since 2015 he takes part in the development of the Versatile Video Coding (VVC) standard by contributing to the work of the Joint Video Exploration Team (JVET).

Adarsh Krishnan Ramasubramonian received the B.Tech. degree from IIT Madras, Chennai, India, in 2006, and the M.S. and Ph.D. degrees from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2008 and 2011, respectively, all in electrical engineering. He has been with Qualcomm Technologies Inc., San Diego, USA, since 2011 and is currently Senior Staff Engineer. He has been an active member of the video coding standardization work by JCT-VC and JVET, and more recently involved in Geometry-based Point Cloud Compression (G-PCC) standardization in MPEG 3DG. His research interests include image/video processing, coding, and communication, HDR content processing and compression of point clouds.

Geert Van der Auwera received the Ph.D. degree in Electrical Engineering from Arizona State University, Tempe, AZ, USA, in 2007, and the Belgian MSEE degree from Vrije Universiteit Brussel (VUB), Brussels, Belgium, in 1997. Presently, he is a Director at Qualcomm Technologies Inc. in San Diego, CA, USA, in the Multimedia R&D & Standards group where he actively contributed to the standardization efforts of JCT-VC's High Efficiency Video Coding (HEVC) and JVET's Versatile Video Coding (VVC). Currently, he is participating in MPEG's Point Cloud Compression activity. Until Jan. 2011, he was with Samsung Electronics in Irvine, CA, USA. Until Dec. 2004, he was Scientific Advisor with IWT-Flanders, the Institute for the Promotion of Innovation by Science and Technology in Flanders, Belgium. In 2000, he joined IWT-Flanders after researching wavelet video coding at IMEC's Electronics and Information Processing Department (VUB-ETRO) in Brussels, Belgium. In 1998, his MSEE thesis on motion estimation in the wavelet domain received the Barco and IBM prizes by the Fund for Scientific Research of Flanders, Belgium. His research interests are video coding, point cloud compression, XR, video traffic and quality characterization, video streaming mechanisms and protocols.