IEIE Transactions on Smart Processing and Computing, vol. 12, no. 2, April 2023
https://doi.org/10.5573/IEIESPC.2023.12.2.122

Overview of Versatile Video Coding (H.266/VVC) and Its Coding Performance Analysis

Minhun Lee[1], HyeonJu Song[2], Jeeyoon Park[3], Byeungwoo Jeon[3], Jungwon Kang[4], Jae-Gon Kim[5], Yung-Lyul Lee[2], Je-Won Kang[6], and Donggyu Sim[1,*]

[1] Department of Computer Engineering, Kwangwoon University / Seoul, Korea {minhun, dgsim}@kw.ac.kr
[2] Department of Computer Engineering, Sejong University / Seoul, Korea hjsong@sju.ac.kr, yllee@sejong.ac.kr
[3] Department of Electrical and Computer Engineering, Sungkyunkwan University / Kyunggi-do, Korea {jiyoonpark, bjeon}@skku.edu
[4] Communication and Media Research Laboratory, Electronics and Telecommunications Research Institute / Deajeon, Korea jungwon@etri.re.kr
[5] School of Electronics and Information Engineering, Korea Aerospace University / Kyunggi-do, Korea jgkim@kau.ac.kr
[6] Department of Electronic and Electrical Engineering and Graduate Program in Smart Factory, Ewha Womans University / Seoul, Korea jewonk@ewha.ac.kr

***** Corresponding Author: Donggyu Sim

Received November 16, 2022; Revised December 21, 2022; Accepted December 31, 2022; Published April 30, 2023

** Regular Paper*

***Abstract*:** Versatile Video Coding (H.266/VVC) is the newest video coding standard jointly developed by the Joint Video Experts Team (JVET), which is organized by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). H.266/VVC provides about 40% bitrate reduction compared with High Efficiency Video Coding (H.265/HEVC) for the same visual quality. This paper introduces in detail the core structure of H.266/VVC by highlighting its features within block partitioning structure, intra/inter prediction, transform, quantization, and in-loop filtering, compared to its predecessor (H.265/HEVC). H.266/VVC yields significantly improved the coding performance, but it increased the computational complexity, particularly for the encoder side, which remains a problem to be tackled for successful commercialization in the future. This paper examined the statistical performance of H.266/VVC coding tools from the bitstreams encoded by the VVC test model (VTM12.0) encoder through rate-distortion optimization under the JVET common test conditions. In addition, the complexity and performance analyses are conducted on the block partitioning structure and group of picture structure. It is expected that an optimized H.266/VVC encoder can be designed and developed by minimizing the coding loss based on the analysis data.

***Keywords*:** Moving picture experts group (MPEG), Joint video experts team (JVET), Video coding experts group (VCEG), Versatile video coding (H.266/VVC), Video compression, Video coding

1. Introduction

Recently, video-based application service industries, such as over-the-top (OTT) media services, video conferencing, and real-time streaming services, have been developed rapidly. As a result, video data accounts for an exponential increase in Internet traffic and consumer demand for more diverse and high-quality video content, such as 360-degree videos for immersive media, screen sharing, and game application services, including 4K ultra high definition (UHD) and 8K UHD video [1, 2]. In order to efficiently serve these various types of large-capacity video data, a new codec capable of providing higher compression efficiency than High Efficiency Video Coding (H.265/HEVC) [3] was required. Accordingly, the international standardization organizations ITU Telecommunication standardization sector (ITU-T) Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) jointly formed Joint Video Experts Team (JVET) with the goal of doubling the compression efficiency compared to H.265/HEVC. It officially began the standardization process of H.266/VVC in July 2020 [4].

H.266/VVC was developed to facilitate efficient

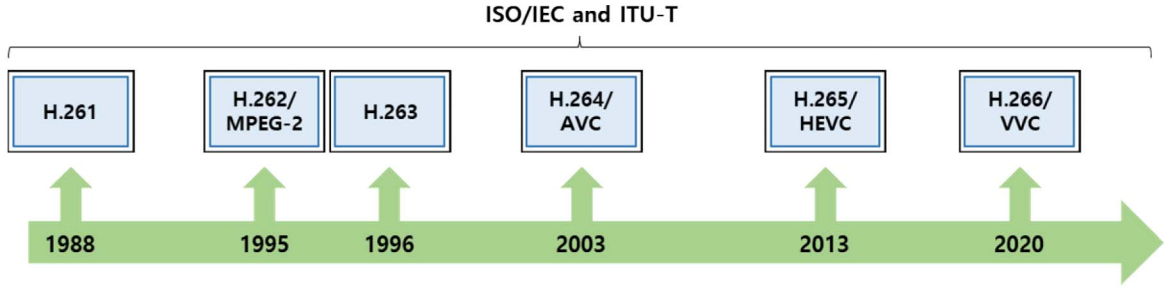


Fig. 1. Development of video coding standards (1988~2022).

compression for a wide range of video content and services, such as high-resolution (up to 8K or higher), high dynamic range/wide color gamut (HDR/WCG) video and screen contents, and 360-degree videos. In addition, H.266/VVC has the same block-based hybrid coding structure as H.264/AVC [5], H.265/HEVC [6], etc., which are conventional video coding standards, and additional techniques to obtain higher coding performance than the existing video coding standards, showing approximately 40% bitrate reduction compared to H.265/HEVC. In H.266/VVC, as in H.265/HEVC, a frame is partitioned into non-overlapping coding tree units (CTUs), a basic coding unit. Furthermore, the maximum CTU size is allowed at 128×128 for efficient higher-resolution video processing. Each CTU was partitioned into only square coding units (CUs) through a recursive quad-tree (QT) split in H.265/HEVC. On the other hand, a multi-type tree (MTT) split consisting of a binary tree (BT) and ternary tree (TT) has been adopted in H.266/VVC, which can be partitioned into rectangular CUs by applying it to QT leaf nodes. H.266/VVC has been developed by improving the predecessor (H.265/HEVC) coding techniques and adopting various new coding techniques to compress various video content effectively. The intra prediction in H.266/VVC allows up to 67 angular prediction modes instead of 35 modes of H.265/HEVC. In addition, pretrained matrix-based and correlation-based prediction between luma and chroma have been newly adopted. For an inter prediction, H.266/VVC supports both the whole block-based (same as H.265/HEVC) and subblock-based prediction techniques. In addition, motion refinement tools have been adopted to use more accurate motion information in a construction scheme for an extended motion vector candidate list. In addition, H.266/VVC also supports performing the transform of the residual signal by explicitly or implicitly selecting from among various transform kernels, and a secondary transform has been adopted to further reduce the redundancy of transform coefficients. Moreover, new in-loop filtering techniques have been introduced into H.266/VVC to use signal ranges better to improve the coding efficiency and reduce the coding artifacts introduced by the quantization and transform process. In addition, various new techniques have been introduced in H.266/VVC, which will be discussed in detail in later sections.

With many new techniques and coding structures, H.266/VVC can obtain comparable subjective quality with H.265/HEVC at only approximately half the bitrate for test sequences. On the other hand, there are some concerns regarding the complexity of the encoder and decoder for practical commercial implementations. Many studies have been conducted to solve this problem, but research on optimization, acceleration, and parallelization is needed because of the difficulties in real-time encoding for the commercialization of H.266/VVC.

Several overview papers for H.266/VVC have been reported [7, 9]. Those papers commonly describe H.266/VVC tools compared to H.265/HEVC, albeit briefly. In addition, each paper focuses on an overview of the first version of H.266/VVC including a comparison against H.265/HEVC [7] and focuses on several interesting consumer electronic use cases and applications [8] aims to explain how new features in H.266/VVC provide the versatility of applications and functionalities [9], respectively. Furthermore, they also present experimental results for BD-rate comparison of H.265/HEVC, and en/decoding complexity for each module in H.266/VVC. This paper provides a detailed description of all the coding tools newly adopted in H.266/VVC. It reports statistical analysis coding techniques for each module of H.266/VVC by analyzing the bitstreams encoded using the VVC test model, VTM12.0 [10]. In particular, the coding efficiency is analyzed according to the CTU size, which is the basic unit of the block-based coding process, and the complexity and impact of BT and TT, which are newly introduced in the block partitioning structure of H.266/VVC, are evaluated. H.266/VVC occupies many complexities in the encoding process. This paper also reviews the performance of adjusting the number of available reference pictures for encoding configuration and provides data for video encoder design and future research by analyzing the ratio selected by the reference structure and video resolution for each coding technology.

The remainder of this paper is organized as follows. Section 2 examines the history of video coding standard technology, and Section 3 examines the newly added technologies adopted in H.266/VVC. Section 4 performs a comparative analysis of the major technologies and structures changed or expanded in H.266/VVC along with a statistical analysis of the above-mentioned various compression techniques of H.266/VVC. This paper is concluded with the outlook in Section 5.

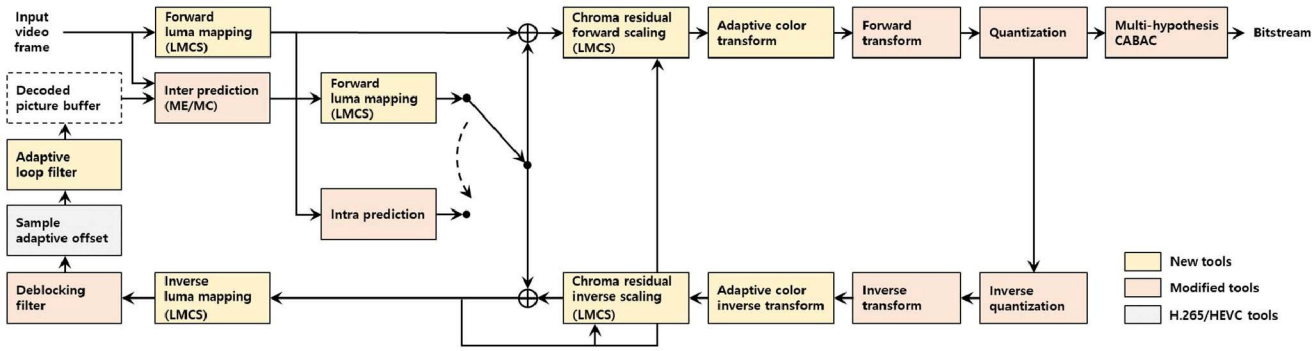


Fig. 2. Block diagram of H.266/VVC encoder.

2. Brief Review of Video Coding Standards

Video compression technology has evolved continuously with technological advances in broadcasting and communication. On the other hand, its commercial adoption on a wide scale would not be possible without affordable chip technology and processing and its high-performance algorithms. The demand for higher video quality and faster communication has called for powerful compression techniques endorsed by well-known and authoritative international standardization bodies, such as ISO/IEC and ITU. Fig. 1 summarizes the development of the recent video coding standards. Although H.261 [11] is straightforward, it can be considered a forerunner of video coding commercially deployed on a wide scale. H.261 was developed by the ITU-T study group (SG)16 and finalized in 1988 for audiovisual services over ISDN, which was under deployment from late 1980 to early 1990. Note that its basic structure of motion compensated prediction and transform coding of prediction residual has been the core structure in later generations of video compression technologies. In 1995 and 1996, H.262 [12] and H.263 [13] were jointly developed by ITU-T SG16 VCEG and ISO/IEC MPEG. H.262, which is more widely referred to as MPEG-2, is a very successful international standard, and it made digital TVs and STBs globally spread at an affordable price. Its main application has been video services of standard definition (SD) and HD quality, mainly over terrestrial, satellite, cable, and ATM. The development has continued to H.264/AVC (also known as H.264/MPEG-4 Part 10) in 2003. H.264/AVC can be noted as the first coding scheme, which employed arithmetic coding for entropy coding of video data in addition to the Huffman coding tool. In addition to Huffman coding, it also employed binary arithmetic coding (BAC) operating in context; thus, it is called context-adaptive binary arithmetic coding (CABAC). The Huffman entropy coding is completely replaced with an arithmetic coder after H.264/AVC. In 2013, H.265/HEVC is released targeting HD and UHD. H.265/HEVC can be noted as introducing very flexible block partitioning structures that implement the idea of variable-block size video coding by enabling a prediction and transformation in variable sizes and shapes [14]. It has an additional transform beyond discrete cosine transform (DCT), i.e., discrete sine transform (DST), which was applied to small intra-predicted luma blocks [15]. Rate-distortion optimized quantization (RDOQ) has been also adopted in H.265/HEVC [16]. Similar to the previous generations of video coding, H.265/HEVC has its extensions, such as scalability extensions, 3D video coding, and fidelity extensions [7]. In 2020, the latest international video coding standard at this point, H.266/VVC, is released by JVET. It targets HD/UHD and higher resolutions as well as 360-degree videos and screen content. Many existing coding tools from H.265/HEVC were redesigned with more flexible structures. For example, the quad-tree block partitioning scheme in H.265/HEVC was replaced with quad-tree plus multi-type tree (QT+MTT) in H.266/VVC, and the number of directions supported in the angular prediction is further increased to sixty-five (65). In addition, H.266/VVC has also adopted various newly developed structures and coding tools. For example, the multiple reference lines (MRL) and intra sub-partitioning (ISP) are introduced to intra prediction. Affine motion compensation (AMC) and adaptive motion vector resolution (AMVR) have been adopted for inter prediction; multiple transform selection (MTS) has been adopted for transform coding; and dependent quantization (DQ) has been newly introduced. H.266/VVC has adopted two tools that are different from its predecessors, i.e., matrix-based intra prediction (MIP) and low-frequency non-separable transform (LFNST), in which their core design is based on training. The predictor-generating matrix in MIP and the transform kernels in LFNST are designed not analytically but through training with numerous sample data.

3. Overview of Versatile Video Coding

Fig. 2 shows a block diagram of H.266/VVC encoder. H.266/VVC uses a block-based hybrid coding structure like conventional video coding standards such as H.264/AVC, H.265/HEVC, etc. The hybrid refers to combining predictive coding and transform coding with the quantization of a residual signal to reduce the spatial and temporal redundancy in the video signal. As shown in Fig. 2, there are tools in H.266/VVC that are the same as or modified from H.265/HEVC. In addition, some techniques have been newly adopted to improve coding efficiency. This section describes the main technique of

H.266/VVC, which shows a compression efficiency of approximately twice that of H.265/HEVC at the same visual quality.

H.266/VVC supports three types of hierarchical temporal prediction structures to ensure efficient compression performance depending on the purpose of compression, same as H.265/HEVC: all-intra (AI), low-delay (LD), and random access (RA) structures. In addition, H.266/VVC specifies two types of intra random access point (IRAP) pictures: instantaneous decoding refresh (IDR) picture, clean random access (CRA) picture, and one type of gradual decoding refresh (GDR) picture [17] for random access. In IDR and CRA pictures having the same concept as H.265/HEVC, the bitrate increases rapidly because the entire picture is encoded using intra prediction. The GDR picture was introduced in H.266/VVC for low-latency applications to alleviate this problem. The area within the GDR picture consists of three areas: intra-coded area, clean (or refreshed) area, and dirty (or non-refreshed) area, and the clean area and dirty area can never propagate errors [18]. Because the clean area of the current GDR picture is reconstructed by referencing the clean area of previous pictures in the GDR period, it is possible to reconstruct the entire area completely without errors occurring in the transmission process, even if the entire area is not intra-predicted. Therefore, H.266/VVC can also be applied to ultra-low latency applications because GDR can smooth out the bitrate of a bitstream, reducing end-to-end latency significantly [19].

In H.266/VVC, each picture can be divided into multiple subpictures that can be en/decoded and transmitted independently. Each picture or subpicture consists of one or more slices with multiple CTUs. In addition, each CTU can be partitioned into CUs by applying the QT+MTT split structure, then performing prediction, transform, and reconstruction. Moreover, H.266/VVC has adopted a CTU dual tree structure where luma and chroma components can have separate coding trees [20].

The prediction of H.266/VVC can be classified mainly into intra prediction and inter prediction [21, 23]. For intra prediction, several techniques have been newly adopted as follows. The 65 directional angular prediction modes, which is approximately doubled in 33 directional angular prediction modes for H.265/HEVC, a wide angle intra prediction (WAIP) mode for rectangular CUs, multiple reference line (MRL) that can perform a prediction with non-adjacent reference lines, and a matrix-based intra prediction (MIP) mode have been adopted. The MIP mode performs a prediction using adjacent reference samples and a predefined matrix obtained from pre-training. Similar to residual quad-tree (RQT) of H.265/HEVC, intra sub-partition (ISP) mode divides the current block into subblocks and performs prediction and transform for each subblock. To improve the coding efficiency of the chroma signal using the luma-chroma correlation, a cross-component linear model (CCLM) mode performs prediction based on a linear model whose parameters are derived from the collocated reconstructed luma samples and reconstructed adjacent chroma samples. In addition, a position dependent prediction combination (PDPC) that generates a final prediction signal by combining the initial intra-predicted samples and adjacent reference samples has been adopted.

In a H.266/VVC inter prediction, motion compensation (MC) is performed by generating a motion vector predictor (MVP) based on the merge or advanced motion vector prediction (AMVP) modes, as in H.265/HEVC. In particular, a pairwise average MVP (PAMVP) has been adopted in H.266/VVC that generates a new motion vector candidate from already constructed motion vector candidates of the merge list to increase the coding efficiency from various motion vector candidates. A history-based MVP (HMVP) is adopted used to increase the coding efficiency, which uses the MV of previously coded blocks as motion vector candidates for the current block. In addition, subblock-based MC methods are newly introduced in H.266/VVC, affine MC (AMC), and subblock-based temporal MVP (SbTMVP) mode. The AMC is a prediction method for performing subblock-based MC through MVs, which are derived from the affine motion model that could represent translational motion and rotation and zoom in/out. Subblock-based MC performs SbTMVP mode through motion information of the corresponding block in the previously decoded picture. Various techniques have been adopted to improve coding efficiency by refining the MVs of the current block. In particular, the merge mode with MV difference (MMVD) mode derives the MVD value with simplified signaling without explicitly transmitting MVD in merge mode. Unlike the MMVD, decoder-side MV refinement (DMVR) mode has been adopted, which applies bilateral matching to refine the accuracy of the initial MV, which be obtained from the signaled merge index. Moreover, the motion or prediction signal of the current block can be refined by performing bi-directional optical flow (BDOF), a technique that refines motion in pixel-based optical flow, and a prediction refinement with optical flow (PROF), which refines the pixel values in a prediction block generated by the AMC. In addition, the techniques of generating a final prediction signal by combining multiple prediction signals have been newly adopted as follows: the combined inter-intra prediction (CIIP) mode, which generates the final prediction signal from the weighted sum of the intra prediction signal and inter prediction signal; the geometric partitioning mode (GPM) mode, which generates the final prediction signal from a weighted sum of two inter prediction signals by applying a mask determined according to the signaled mode; the bi-directional prediction with CU weights (BCW), which uses the signaled weights to generate a final prediction signal from a weighted summation of the two inter prediction signal.

Furthermore, techniques for reducing the bits of MVD to be transmitted have been also adopted, the adaptive motion vector resolution (AMVR) and symmetric MVD (SMVD). The AMVR mode can reduce the transmission bits of MVD by adaptively changing the resolution of MVD. The SMVD mode signals MVD only for one reference list and derives MVD for the other by assuming linear motion. Various techniques in the transform and quantization process for coding efficiency have been

adopted in H.266/VVC [24, 25].

In H.265/HEVC, the transform using the DCT-II kernel was applied to each square transform unit (TU), while the transform using the DST-VII kernel is only applied to the 4×4 intra prediction block. H.266/VVC supports transforms using vertical and horizontal transform kernels of different types and lengths for rectangular transform blocks (TBs) because of the different shapes of blocks. In particular, multiple transform selection (MTS) has been adopted in H.266/VVC to transform using DCT-II, DST-VII, and DCT-VIII kernels based on the prediction mode or explicit signaling. The low-frequency non-separable transform (LFNST) has been also adopted, which can achieve an additional coding efficiency by applying the secondary transform to the low-frequency coefficients after the transform for the intra predicted block. In addition, subblock-based transform (SBT) has been adopted, transforming only on subblocks for the inter predicted block according to the signaling index. For the quantization process of H.266/VVC, a dependent quantization (DQ) and joint coding of chroma residual (JCCR) have been adopted. The DQ uses two quantizers and performs the transition between the two quantizers according to the transform coefficient level. The JCCR is used when the quantized chroma residual signals are similar and transmit only a single chroma residual signal. The entropy coding process of H.266/VVC is performed through context-based adaptive binary arithmetical coding (CABAC) [25]. The CABAC of H.266/VVC uses the same algorithm as H.265/HEVC but uses a multi-hypothesis probability estimation model instead of a look-up table to increase the accuracy of probability estimation.

The in-loop filtering process of H.266/VVC specifies the adaptive loop filter (ALF), cross-component ALF (CC-ALF), and the luma mapping chroma sampling (LMCS) in addition to deblocking filter (DF) and sample adaptive offset (SAO) used in H.265/HEVC [26]. The LMCS is a technique to reflect the characteristics of HDR videos. It aims at using the signal range better for improved coding efficiency rather than specifically addressing coding artifact reduction. The ALF and CC-ALF are the third filtering process in the decoding process of H.266/VVC, which performs block-based linear filtering and adaptive clipping with the filter coefficients determined to minimize the reconstruction error.

As previously mentioned, H.266/VVC uses the hybrid coding structure, and various techniques have been modified or newly adopted compared to H.265/HEVC. The subsequent subsection describes the newly introduced or modified tools compared to the predecessor (H.265/HEVC) within each module. Table 33 is provided with notations used in Section 3.

3.1 Structure of Picture/Block Partitioning

The video sequence is composed of multiple pictures, and each picture could be divided hierarchically and processed for coding efficiency and parallel processing. In H.264/AVC, the macroblocks have been used as a basic unit of compression, which is fixed to 16×16 [5]. In

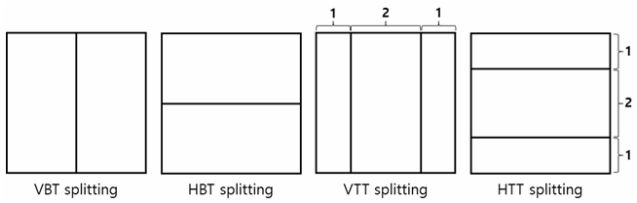


Fig. 3. An example of MTT split types.

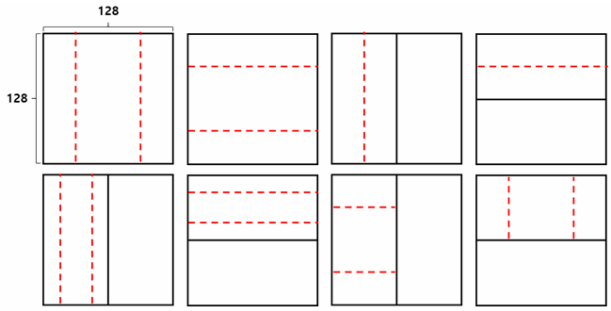


Fig. 4. Disallowed BT and TT split for 64×64 VPDU constraints.

H.265/HEVC, each picture could be divided into one or several tiles and slices, which are a sequence of CTUs. Subsequently, a recursive partition is performed on each CTU with a QT split structure for functional subdivision while having more sizes than its predecessor (H.264/AVC), and the leaf node is called CU. Thereafter, the CU is divided into a prediction unit (PU), which is a basic unit of prediction, and a TU, which is a basic unit of transform, to perform compression efficiently [14].

As mentioned earlier, in H.266/VVC, a picture can be divided into one or several tiles and slices, which have similar concepts to H.265/HEVC, and subpictures consisting of one or more rectangular slices. In addition, H.266/VVC can use a CTU of up to 128×128, which is four times larger than H.265/HEVC, as a basic unit of compression to process video with a larger resolution than H.265/HEVC. In addition, each CTU can be first divided through the QT split, and the QT leaf node can then be partitioned further through the MTT split for more flexible partitioning. As shown in Fig. 3, the MTT split has four splitting types: vertical binary tree (VBT) split, horizontal binary tree (HBT) split, vertical ternary tree (VTT) split, and horizontal ternary tree (HTT) split. The QT+MTT leaf node is a CU, and compression can be performed using a rectangular CU in H.266/VVC. Unlike H.265/HEVC, the block sizes of CU, PU, and TU are the same except when CU is larger than the maximum TU size, when predicting in ISP mode, and performing a transform through SBT mode. In addition, H.266/VVC has adopted the GPM mode, which performs the prediction and reconstruction through a more flexible non-rectangular partition rather than a vertical and horizontal rectangular partition, showing improved coding efficiency. In addition, the CTU dual tree is adopted in H.266/VVC, which allows the luma and chroma components to use separate partitioning structures. This can only be applied to an intra-slice, and the compression efficiency can be improved because luma

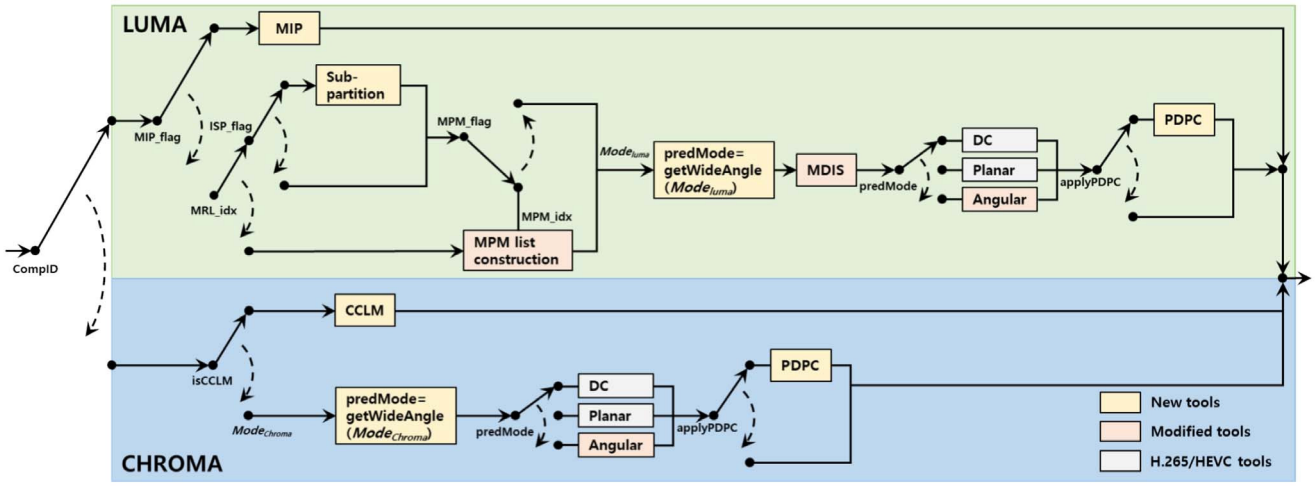


Fig. 5. Block diagram of intra prediction in H.266/VVC decoder.

and chroma components are divided to suit each characteristic. In the case of inter-slice, luma and chroma components are divided into the same QT+MTT split structure [20].

When designing and implementing a hardware video decoder, major coding techniques are divided into functions and correlations, and the pipelines are configured to enable parallel processing. The maximum block size of the pipeline is determined by the TB size, and it is possible to design to operate by dividing the block to be processed into arbitrary small blocks. For this reason, a virtual pipeline data unit (VPDU) has been introduced in H.266/VVC, like H.265/HEVC. The size of the VPDU is 64×64 because the maximum TB size in H.266/VVC uses 64×64, or it is limited to the size of the maximum CU if the size of the maximum CU is smaller than 64×64. In addition, because different consecutive VPDUs in the hardware video decoder are processed simultaneously in the pipeline stage, the constraint is defined not to allow specific segmentation for CUs with a width or height of 128 for CUs larger than the size of VPDUs, as shown in Fig. 4.

3.2 Intra Prediction

This section describes the main intra prediction techniques of H.266/VVC, as shown in Fig. 5. In H.265/HEVC, intra prediction is performed using 33 angular prediction modes, planar mode, and DC mode, while in H.266/VVC, various techniques have been adopted to achieve improved coding efficiency, resulting in better prediction accuracy by performing prediction using 65 angular prediction modes, wide-angle intra prediction (WAIP) mode, multiple reference line (MRL), intra sub-partition (ISP) mode, matrix-based intra prediction (MIP) mode, planar mode, DC mode, position dependent prediction combination (PDPC) mode, and cross component linear model (CCLM) mode [21].

Angular prediction modes with 65 angles and wide angle intra prediction (WAIP): H.266/VVC basically supports 65 angular prediction modes, which is increased by approximately two times to perform a precise prediction for CUs with a larger size than H.265/HEVC.

Table 1. Angular prediction modes are replaced by WAIP mode depending on different aspect ratios.

Aspect ratio (W:H)	Angular prediction mode (to be replaced)	Replaced WAIP mode
16:1	2 to 15	67 to 80
8:1	2 to 13	67 to 78
4:1	2 to 11	67 to 76
2:1	2 to 7	67 to 72
1:1	None	None
1:2	61 to 66	−6 to −1
1:4	57 to 66	−10 to −1
1:8	55 to 66	−12 to −1
1:16	53 to 66	−14 to −1

In general, the same number of angular prediction modes are assigned to the upper and left sides of the CU to perform prediction for a square CU. On the other hand, H.266/VVC, compression is performed using a square and rectangular CU. WAIP has been adopted, which allocates more angular prediction modes to the size with greater width or height, to perform a sufficient prediction using angular prediction mode for rectangular CU and ensure improved prediction accuracy. Because WAIP mode is used without additional signaling, some angular prediction modes are replaced by 180-degree rotated modes depending on the aspect ratio of the CU, as shown in Table 1.

Reference sample filtering: Similar to H.265/HEVC, two filtering techniques, reference sample smoothing, and interpolation filtering, are applied to reference samples for an intra prediction of H.266/VVC. The reference samples may have a discontinuity because block-based prediction and reconstruction are performed. In addition, the reference samples located at the fractional-sample position must be generated to perform a prediction in fractional-slope angular modes. This can be addressed by performing the filtering process on the reference samples as follows:

for the reference sample smoothing, the reference sample is filtered with the finite impulse response filter {1, 2, 1}/4. Furthermore, reference sample smoothing is performed for integer-slope angular modes if the number of samples in the current block is more than 32. Interpolation filtering is applied to the reference samples around the fractional-sample position with a DCT-based interpolation filter (DCTIF) for fractional-slope angular modes. The DCTIF is constructed in the same way as the chroma DCTIF used for motion compensation in both H.265/HEVC and H.266/VVC [27]. For the luma block, 4-tap interpolation filters are used for reference sample filtering, and the linear 2-tap interpolation filter of H.265/HEVC is used in H.266/VVC for the chroma components.

Most probable mode (MPM): In H.266/VVC, the MPM list is constructed based on left and above neighbor CUs and used to reduce the amount of transmitted bits for signaling prediction modes such as H.265/HEVC. On the other hand, unlike H.265/HEVC, an MPM list is constructed with 6 MPMs, including planar mode, which is signaled with a separate flag. The prediction mode is considered planar mode if the prediction mode of the neighboring CU cannot be referenced or predicted with MIP mode. The MPM list is constructed based on four cases according to the prediction modes of the neighboring blocks to signal the prediction mode.

Position dependent prediction combination (PDPC): In H.266/VVC, because intra prediction is performed using the reference samples above and left of the CU, the prediction accuracy may be lower for prediction samples at a location far from the reference samples, such as at the bottom right of the CU. H.266/VVC has adopted the PDPC to address this problem. The PDPC generates final prediction samples by performing a linear combination using adjacent reference samples that are filtered or not based on the CU size and the intra-prediction mode, and the initial intra-predicted samples.

Multiple reference line (MRL): In H.264/AVC and H.265/HEVC, intra prediction is performed using the above and left nearest reference line of the CU based on the correlation of the adjacent pixels. On the other hand, because the reference samples in the reference line are reconstructed samples, not original, compression error may damage the adjacent reference samples, which may be unsuitable for prediction. To compensate for this problem, H.266/VVC uses reference samples in two non-adjacent reference lines in addition to the adjacent reference line. The index for the reference line is signaled, and when a prediction is performed using non-adjacent reference lines, only the MPM-based prediction, excluding planar mode, can be used, and the PDPC cannot be applied.

Intra sub-partition (ISP): In general, a prediction is performed in a video compression process, and a transform process is performed on a residual signal that is a difference value between the original signal and the prediction signal. In this case, the compression error may spread to the entire block if the transform is performed at once if the residual signal in the residual block is distributed biased to one side. In addition, even if the directionality within that block is the same for a relatively large block, the prediction sample with a distance from the reference sample may be inaccurate if the prediction is performed at once. In this case, the accuracy of the prediction may be improved if the prediction is performed by dividing into multiple small CUs, but there is a problem that the bits for partitioning increase accordingly. ISP mode has been adopted for H.266/VVC to solve this problem. ISP mode cannot be applied to a 4×4 CU. Moreover, if the ISP mode is applied, prediction and transform processes are performed by dividing the PU or TU into two or four vertically or horizontally according to the aspect ratio of the CU, as shown in Table 2. At this time, sub-partitions with a width less than 4, such as 1×N and 2×N, could constitute an issue for hardware implementation. Therefore, a prediction is performed at once using only the adjacent reference samples to 4×N regions grouped into these sub-partitions. The corresponding four 1×N and 2×N TBs perform the transform in parallel [21, 28]. In addition, the same prediction mode is applied for each PU, and the ISP mode is applied only when the MRL index is set to 0, and all ISP sub-partitions could be conducted PDPC in the same way, as in the non-ISP case.

Matrix-based intra prediction (MIP): MIP mode is a newly adopted intra prediction mode for H.266/VVC. It was initially proposed as a neural network-based prediction method. However, a set of pre-trained matrices for three cases according to the CU size is defined as a table and simplified to a signaling index to improve computational complexity, as shown in Table 3 [29]. The prediction process of the MIP mode is performed following three steps, as shown in Fig. 6. First, averaging

Table 2. Number of sub-partitions of the prediction and transform for ISP mode.

CU size	Hor. split		Ver. split	
	Num. of Pred. sub-part.	Num. of Trans. sub-part.	Num. of Pred. sub-part.	Num. of Trans. sub-part.
4×8	2	2	1	2
8×4	2	2	2	2
4×N (N>8)	4	4	1	4
8×N (N>4)	4	4	2	4
Others	4	4	4	4

Table 3. MIP cases depending on CU size.

MIP Cases	CU size
Case 1	4×4
Case 2	(W == 4 or H == 4) or 8×8
Case 3	Others

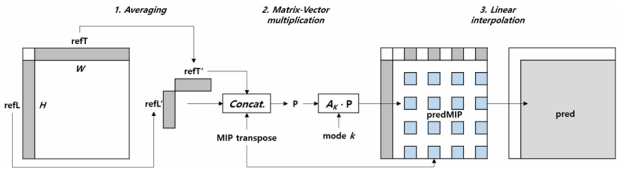


Fig. 6. Flowchart of MIP process [21].

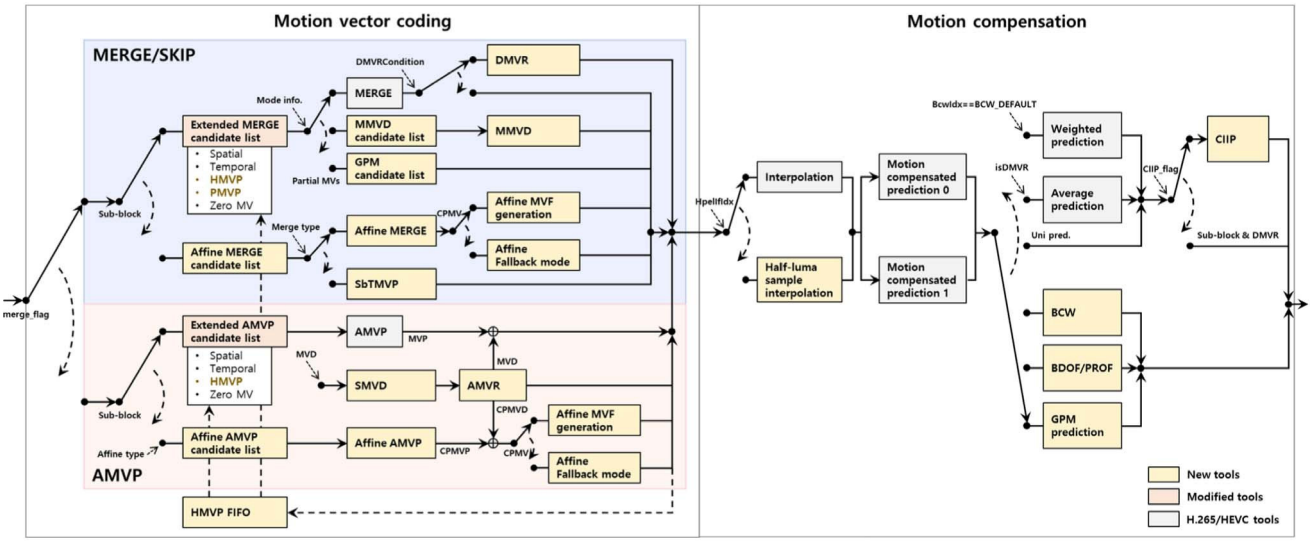


Fig. 7. Block diagram of inter prediction in H.266/VVC decoder.

is performed to the above and left reconstructed reference samples, refT and refL, to obtain the reduced smaller boundaries, refT' and refL'. In the second step, the predMIP signal is generated by matrix-vector multiplication through the vector p, which is concatenated with refT' and refL' in the first step, and the matrix A_k, which is determined by the signaled index. In the final step, the remaining samples in PU are derived by linear interpolation, where horizontal interpolation is first performed, and vertical interpolation is then performed.

Cross component linear model (CCLM): Generally, when compressing the YUV color format video, the luma component is first compressed, followed by the chroma component is compressed. Since video signal generally has local dependency between different color components, H.266/VVC uses this property by adopting CCLM mode, a prediction technique to improve the coding efficiency by minimizing the regression error between the reconstructed luma component and the reconstructed chroma component. Since a lot of computational complexity is consumed in the process of obtaining a linear model with all reference samples, it has been simplified to obtain a linear model using reference samples at pre-determined positions according to each CCLM mode [30].

3.3 Inter Prediction

As shown in Fig. 7, H.266/VVC has adopted various inter prediction techniques. The coding efficiency heavily relies on the efficient representation of motion information. Efficient motion data coding is realized by AMVP mode that predicts the MV values using a list of predictors and merge/skip modes that derive the complete motion information based on the neighboring motion data. Merge and AMVP modes in H.266/VVC are an extension of those in H.265/HEVC to increase the prediction accuracy while minimizing the bitrate. In merge modes of H.266/VVC, there are block-based merge modes consisting of general merge mode, merge with motion vector difference (MMVD) mode, combined inter and intra prediction (CIIP) mode, geometric partitioning mode (GPM), and subblock-based merge mode consisting of subblock-based temporal motion vector prediction (SbTMVP) and affine motion model-based motion compensation prediction. Inter-prediction modes in skip mode, where residuals are not signaled and inferred to be zero, are the same as the merge modes, excluding CIIP mode. AMVP modes in H.266/VVC consist of general inter mode (regular AMVP mode), symmetric MVD (SMVD) mode, and affine inter mode. In addition, H.266/VVC has adopted decoder-side motion vector refinement techniques, which are decoder-side MV refinement (DMVR), bi-directional optical flow (BDOF), and a prediction refinement with optical flow (PROF) to improve the prediction accuracy without signaling additional information.

3.3.1 Merge Mode Inter Prediction

Regular merge mode: A merge candidate list consists of newly introduced history-based motion vector prediction (HMVP) and pairwise average MVP (PAMVP) following spatial merge candidates and temporal merge candidates similar to those in H.265/HEVC. HMVP candidates are the MVs of previously coded CUs mostly from non-adjacent CUs, that are stored using a five-entry table and updated using a first-in-first-out (FIFO) rule. The PAMVP is generated by averaging the MVs of the first and second candidates in the merge candidate list for each reference picture list, even if they have different reference pictures, and assigning the reference picture index of the first candidate to that of the PAMVP. If there is only one MV available, it is used directly. If no MV is available for one of the reference picture lists, the MV for the list is considered invalid.

Merge with motion vector difference (MMVD): In MMVD, one of the first two existing candidates in the merge candidate list is selected as the base motion. The base motion is refined with an MVD obtained from a predefined direction and a predefined distance. A

predefined direction indicating the direction of MVD can be either horizontal or vertical direction, resulting in four directions. A direction index is signaled to indicate the selected MVD direction, as shown in Table 4. A predefined distance indicates a distance to be refined from the base motion. There are two predefined distance tables, as shown in Table 5. Based on one of the tables selected at the picture level and a signaled distance index for MVD, the distance from the base motion can be derived for the refined motion. MMVD improves the accuracy of MV from the general merge mode, even though it cannot provide MVs as accurate as those in AMVP mode.

Combined inter and intra prediction (CIIP): Similar to the bi-prediction, i.e., a linear superposition of two motion compensated predictions that can further reduce the energy of the prediction error than uni-prediction, a uni-directional inter prediction can be superposed with intra prediction. In CIIP mode, the predicted signal, P_{CIIP}, is generated by a weighted sum of inter-predicted signals using merge mode, P_{inter}, and intra-predicted signals using the planar mode, P_{intra}, as described in Eq. (3.3.1). The weights, a weight for inter prediction, W_{inter}, and weight for intra prediction, W_{intra}, are derived based on whether the above and left neighboring CUs are coded using intra or inter prediction mode. When both the above and left neighboring CUs are intra-coded or inter-coded (W_{inter}, W_{intra}) is set to (1, 3) or (3, 1), respectively. Furthermore, it is set to (2, 2) if only one of both neighboring CUs is intra-coded.

$$P_{CIIP} = \left(W_{inter} \times P_{inter} + W_{intra} \times P_{intra} + 2\right) \gg 2 \qquad (3.3.1)$$

Geometric partitioning mode (GPM): H.266/VVC does not employ geometric block partitioning that could increase the partitioning precision for moving objects boundaries because of implementation complexity. Instead, GPM mode has been adopted to provide a similar effect to geometric block partitioning. In GPM mode, a CU is partitioned into two parts using a straight line. The straight line is parametrized by an angle and an offset so that there are 64 partitions for a CU with size $w \times h = 2^m \times 2^n$ with $m, n \in \{3, \ldots, 6\}$ excluding 8×64 and 64×8. In each partition, its own MV is derived from performing a block-based motion compensation. The final prediction for the CU is generated by performing a blending process with adaptive weights based on the position of each sample relative to the geometry partitioning boundary, as shown in Fig. 8. A, MV for each partition is uni-directional, so the complexity of motion compensation for the CU is the same as that of bi-directional prediction. The MV for each partition is derived from the regular merge list, as shown in Fig. 9, with the signaled GPM merge index.

Subblock-based merge mode: In subblock-based merge mode, a CU with a height and width larger than or equal to eight luma samples is divided into 8×8 subblocks, and a different MV is derived for each subblock. A subblock-based merge mode consists of SbTMVP similar to TMVP in H.265/HEVC and motion vector prediction

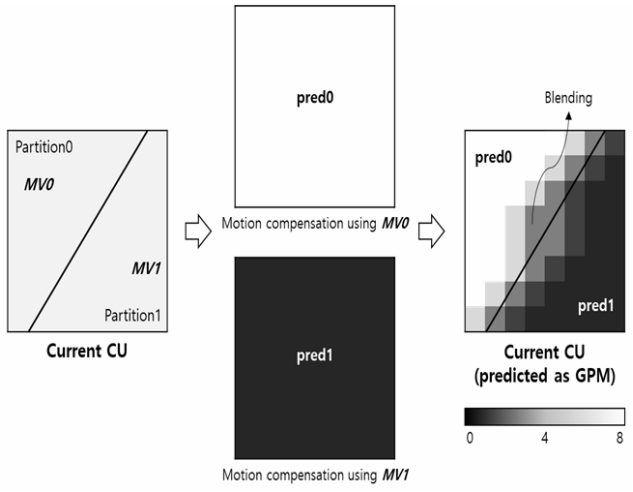


Fig. 8. Derivation of predicted signal using GPM.

Merge Idx	L0	L1
0	A0	A1
1	B0	B1
2	C0	C1
3	D0	D1
4	E0	E1
5	F0	F1

Regular merge list

GPM merge Idx	L0 or L1
0	A0
1	B1
2	C0
3	D1
4	E0
5	F1

GPM merge list

Fig. 9. GPM merge list construction.

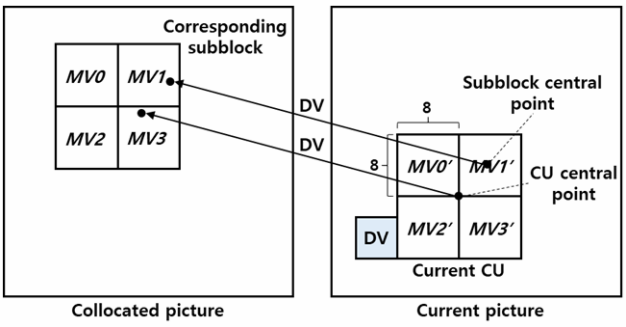


Fig. 10. Derivation of SbTMVP [22].

based on an affine motion model. SbTMVP is only applied to the subblock-based merge mode, and an affine motion model-based prediction is applied to the merge and AMVP modes. Section 3.3.3 describes the affine motion model-based prediction. Fig. 10 presents the derivation process of SbTMVP consisting of two steps: 1) derivation of a displacement vector (DV), and 2) derivation of motion information for each subblock based on the motion information derived by the DV. If the MV of the left-bottom neighboring block refers to the collocated picture, that MV is used as the DV. Otherwise, zero MV is used as the DV. The DV is applied to the central position of the current CU to locate the corresponding sample position in the collocated picture. the SbTMVP is considered available if the block containing the corresponding sample position in the collocated picture is inter-coded. If

available, the motion information of the corresponding subblocks in the collocated picture is found by applying the DV to each subblock in the current CU. Finally, the motion vector of each subblock in the current CU is derived from the motion information of the corresponding subblock, similar to the TMVP process in H.265/HEVC, where temporal motion scaling is applied to align the reference pictures of the temporal motion vectors to those of the current CU [31]. The derived motion vectors for subblocks in the current CU become the SbTMVP.

3.3.2 AMVP Mode Inter Prediction

In the AMVP mode, the components of an MV are coded differentially using an MVP and MV difference (MVD). In H.266/VVC, the AMVP mode is extended using improved predictors, providing a more flexible MVD signaling to improve the tradeoff between the motion accuracy and overhead motion bits. These enhancements on MV coding and motion compensation, including the revised AMVP candidate list construction, AMVR, BCW, and SMVD, are described in the following.

General AMVP mode: The MV prediction algorithm of H.266/VVC is based on the AMVP of H.265/HEVC. The AMVP introduced in H.265/HEVC explicitly signals one of the two potential MVP candidates derived from five spatially neighboring and two temporally co-located MVs. In this way, the motion information, including reference picture indices, MVP indices, and MVDs for each reference picture list 0 and list 1, are singled in the AMVP mode. As a new feature, H.266/VVC adds an MV prediction of HMVP in the merge mode and AMVP candidate list. The HMVP allows the reuse of MVs of previously coded non-adjacent CUs. In addition, the AMVP candidate list construction process is revised in terms of complexity.

Adaptive motion vector resolution (AMVR): H.266/VVC increases the MV precision to 1/16 luma sample, while the HEVC uses only a quarter-luma-sample precision. On top of the higher precision MV representations, a CU-level AMVR method is applied to customize the balance between quality and the MV bit cost overhead. For a CU with translational motion in AMVP mode, MVDs can be coded in units of a quarter, half, integer, or four luma samples. For the affine AMVP mode, MVDs can be switched among the quarter, integer, or 1/16 luma samples. An alternative six-tap smoothing interpolation filter (IF) is used instead of the eight-tap IF from HEVC when a half-luma-sample MV accuracy is used in AMVP mode.

The MVP is rounded to the indicated precision before being added together with the MVD to ensure the reconstructed MV uses the same precision as the MVD. The CU-level MV resolution indication is conditionally signaled if the current CU has at least one non-zero MVD component. Quarter-luma-sample MVD resolution is inferred if all MVD components (i.e., horizontal and vertical MVDs for reference lists 0 and 1, respectively) are zero.

Bi-prediction with CU-level weights (BCW): BCW provides the weighted averaging of the two prediction signals for bi-prediction at the CU-level, in addition to the traditional weighted prediction (WP) for which the weights are specified at the slice level for each reference picture. In H.266/VVC, the legacy explicit-weighted prediction scheme is kept and extended with CU-level syntax control for the weighted averaging. Five weights are predefined, $w \in \{-2, 3, 4, 5, 10\}$, and an index (denoted as $wIdx$) is signaled at the CU level to specify the selected weight w of the prediction block from list 1. All five weights are used when all the reference pictures are temporally preceding the current picture in display order. Otherwise, only the weights $w \in \{3, 4, 5\}$ are used.

Each luma/chroma prediction sample of BCW is calculated as follows:

$$P_{bi-pred} = \left((8-w)\times P_{L0} + w\times P_{L1} + 4\right) \gg 3 \qquad (3.3.2)$$

where $P_{bi-pred}$ is the final prediction and P_{L0} and P_{L1} are prediction samples pointed to by the MVs from reference picture lists 0 and 1, respectively.

BCW is only applied to CUs with a CU size larger than or equal to 256 luma samples. To avoid interactions between WP and BCW, if CU uses WP, then the $wIdx$ is not signaled, and w is inferred to be 4 (i.e., equal weight is applied). For regular merge mode or affine merge mode, the $wIdx$ is inferred from neighboring blocks based on the merge candidate index. CIIP and BCW cannot be jointly applied for a CU. When a CU is coded with CIIP mode, the $wIdx$ is set to 2, e.g., equal weight. The DMVR and BDOF are both turned off when the weight is non-equal.

Symmetric motion vector difference (SMVD): When the motion of the current block is on a constant trajectory over a past and future reference picture in display order, the corresponding MVs and reference picture indices tend to be symmetrical. SMVD exploits this assumption of linear motion to save bits for MVDs and reference picture index signaling in the true bi-direction mode that uses past and future reference pictures, as shown in Fig. 11. When SMVD is applied for a CU, only the MVP indices of lists 0 and 1 and the MVD for list 0 are signaled. Other motion information is derived at the decoder side without signaling. That is, first, the MVD for list 1 $(mvdx_{L1}, mvdy_{L1})$ is set to the reverse of the list 0 MVD $(-mvdx_{L0}, -mvdy_{L0})$, as shown below:

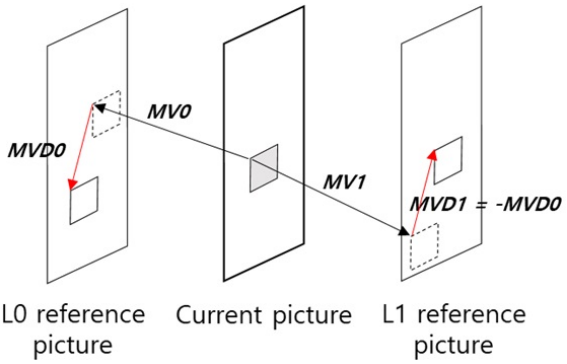


Fig. 11. Illustration for SMVD mode.

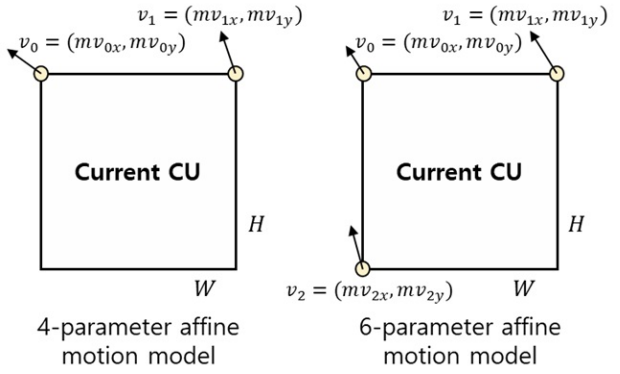


Fig. 12. CPMV based affine motion model.

$$(mvdx_{L1}, mvdy_{L1}) = (-mvdx_{L0}, -mvdy_{L0}) \quad (3.3.3)$$

Second, the lists 0 and 1 reference picture indices are implicitly derived at the slice level. That is, each reference picture is the nearest picture among all pictures in its list, and they have opposite directions to each other.

3.3.3 Affine Motion Compensation

Because H.265/HEVC only considered the translational motion model for motion compensation prediction, it is inefficient when there is motion, such as zoom-in/-out, rotation, or perspective motions. H.266/VVC has adopted the 4-/6-parameter affine motion model for motion compensation prediction to improve the coding efficiency, especially for those cases.

Affine motion models based on control point MV (CPMV), 4-parameter affine motion model using two CPMVs, and 6-parameter affine motion model using three CPMVs, are described in Fig. 12. In the case of the 4-parameter affine motion model using 2 CPMVs which are v_0 and v_1, an MV (mv_x, mv_y) at sample location (x, y) in a CU whose size of $W \times H$ is derived as follows:

$$\begin{cases} mv_x = \dfrac{mv_{1x} - mv_{0x}}{W} x + \dfrac{mv_{1y} - mv_{0y}}{W} y + mv_{0x} \\ mv_y = \dfrac{mv_{1y} - mv_{0y}}{W} x + \dfrac{mv_{1x} - mv_{0x}}{W} y + mv_{0y} \end{cases} \quad (3.3.4)$$

In the case of a 6-parameter affine motion model using three CPMVs, which are v_0, v_1, and v_0 and v_2, an MV (mv_x, mv_y) at sample location (x, y) in a CU whose size of $W \times H$ is derived as follows:

$$\begin{cases} mv_x = \dfrac{mv_{1x} - mv_{0x}}{W} x + \dfrac{mv_{2x} - mv_{0x}}{H} y + mv_{0x} \\ mv_y = \dfrac{mv_{1y} - mv_{0y}}{W} x + \dfrac{mv_{2y} - mv_{0y}}{H} y + mv_{0y} \end{cases} \quad (3.3.5)$$

As expressed in Eqs. (3.3.4) and (3.3.5), an MV at every sample location in a CU can be derived with an affine motion model using CPMVs. On the other hand, to reduce complexity, an MV is derived for each 4×4 subblock in a CU, and motion compensation is performed with the derived 4×4 block level MV, which is an MV at the center sample position of the 4×4 block.

The affine merge mode and affine inter mode are based on the affine motion model in H.266/VVC depending on the ways of obtaining CPMVs. In the affine merge mode, which is a part of the subblock-based merge mode, the affine merge candidates, i.e., CPMV candidates, are added to the subblock-based merge candidate list so that the motion compensated prediction can be performed based on the affine motion model with the derived CPMVs. There are two types of CPMV candidates in affine merge mode: inherited affine merge candidates and constructed affine merge candidates. The inherited affine merge candidates are the CPMVs of the current CU derived from the CPMVs of the above or left neighboring CU, which is motion compensated based on the affine motion model. The constructed affine merge candidates are constructed by combining the translational motion information of neighboring CUs corresponding to the control points.

The affine inter mode, which is a type of AMVP mode, requires information such as a flag to indicate the motion model type, i.e., the 4-parameter model or 6-parameter model, and two or three motion vector differences (MVDs) between the CPMVs and the predictors from an affine MVP list. For up to two affine AMVP candidates for an affine MVP list, there are the inherited affine AMVP candidates and the constructed affine AMVP candidates similar to the affine merge mode. The inherited affine AMVP candidates are derived from the CPMVs of the above or left CU having the same reference picture to the current CU, and the constructed affine AMVP candidates are derived by combining the translation motion information of the neighboring CUs having the same reference picture to the current CU. In addition, if there are fewer than two affine AMVP candidates, the same translational MV of a neighboring CU is assigned to two or three CPMVs in the affine AMVP list, depending on the motion model type.

3.3.4 Decoder-side MV Refinement Tools

Refinements of motion and prediction at the decoder side are introduced to improve the prediction quality without increasing the bit overhead of the signaling motion parameters. DMVR is used to improve the accuracy of the MVs of the regular merge mode with a low-complexity motion refinement. Unlike block-based motion compensation (MC), optical flow is expected to achieve the effect of sample-wise inter prediction. It is implemented in H.266/VVC as BDOF to improve the bi-prediction efficiency and as PROF to refine the subblock prediction of the affine MC (AMC).

Decoder-side motion vector refinement (DMVR): DMVR refines the bi-prediction motion of the regular merge mode using a bilateral search. To ensure the bilateral search with equal distance, DMVR is allowed only if the merge MV pair point to two reference pictures that have an equal and opposite temporal distance to the current picture. As shown in Fig. 13, DMVR applies

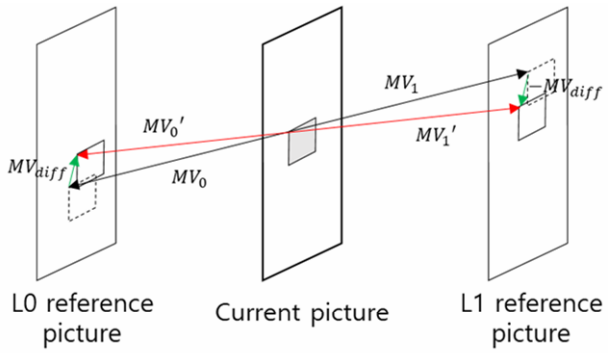


Fig. 13. Bilateral matching based DVMR.

bilateral matching to refine the accuracy of the input MV pair $\{MV_0, MV_1\}$. That is, it searches the candidate MVs around the initial MVs in lists 0 and 1 with a mirrored MV offset MV_{diff}. The refined pair obtained by Eq. (3.3.6), $\{MV'_0, MV'_1\}$, are used for the motion-compensated prediction of both luma and chroma CBs of a CU.

$$MV_0' = MV_0 + MV_{diff}$$
$$MV_1' = MV_1 - MV_{diff} \quad (3.3.6)$$

The searching process consists of an integer sample MV offset search and a fractional sample MV offset refinement. The integer sample MV search calculates the sums of the absolute differences (SADs) between each pair of candidate reference blocks in lists 0 and 1 within the search range of ±2 integer luma samples from the initial MVs. The fractional sample refinement is derived using a parametric error surface approximation instead of additional SAD comparisons.

Bi-directional optical flow (BDOF): BDOF is another coding tool for improving the bi-prediction signal using a motion refinement performed by the decoder. In particular, BDOF aims at compensating the sample-wise fine motion that is limited in the block-based MC based on the optical flow concept at the 4×4 subblock level. It is applied to CUs coded either in merge mode or AMVP mode and assumes constant motion trajectory. As the same constraint applied to DMVR, BDOF is applied only if the two different reference pictures have an equal distance in picture order count (POC) to the current picture.

For each 4×4 subblock, a motion difference relative to CU MVs is calculated by solving an optical flow equation that minimizes the difference between the prediction subblocks of lists 0 and 1. The derived motion differences and the prediction sample gradients are then used to adjust the bi-predicted sample values.

Let $I(i,j,t)$ be the luminance value of a sample at time t in position (i,j). Assuming the luminance of a sample does not change during the object motion, the optical flow equation can be expressed as follows:

$$0 = \frac{\partial I}{\partial t} + v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} \quad (3.3.7)$$

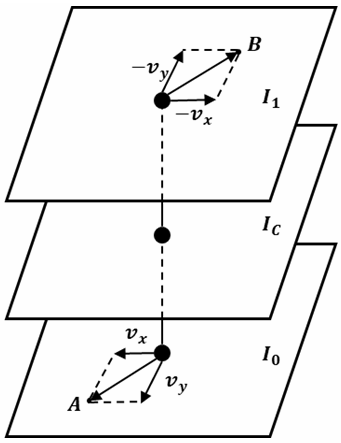


Fig. 14. BDOF using a symmetric motion model [22].

where the motion (v_x, v_y) describing the remaining motion applied on top of the original MV at each sample position.

As shown in Fig. 14, the motion (v_x, v_y) from I_c to I_0 is symmetrical to its motion from I_c to I_1, where I_c, I_0 and I_1 are arrays of luminance values in the current block and the two prediction blocks from the lists 0 and 1 reference pictures, respectively. According to the aforementioned constraint that BDOF is applied only to a true bi-directional prediction with the same prediction distance, the remaining motions relative to both reference pictures are assumed to be in a mirroring relation.

In such a symmetric motion model illustrated in Fig. 14, each sample in I_c can be approximated from two directions, one from its correspondence A in I_0 and the other from its correspondence B in I_1 using Eq. (3.3.7). By minimizing the difference between two predictions with refined motion, the value of (v_x, v_y) is calculated as

$$\min_{(v_x,v_y)} \sum_{(i,j)\in\Omega} \Delta^2(i,j) \quad (3.3.8)$$

$$\Delta(i,j) = I_0(i,j) - I_1(i,j) + v_x\left(\frac{\partial I_0(i,j)}{\partial x} + \frac{\partial I_1(i,j)}{\partial x}\right) + v_y\left(\frac{\partial I_0(i,j)}{\partial y} + \frac{\partial I_1(i,j)}{\partial y}\right) \quad (3.3.9)$$

The vector (v_x, v_y) of each 4×4 subblock is calculated from the extended 6×6 window (denoted as Ω) containing a subblock in the center, assuming that it is constant in each subblock. This way, a more stable motion field is derived with reduced computational complexity. The optimization problem in Eq. (3.3.8) can be solved using the auto- and cross-correlation of the horizontal and vertical gradients for each prediction sample [22].

Based on the derived motion refinement (v_x, v_y) and the prediction sample gradients, the following adjustment is calculated for each sample in the subblock:

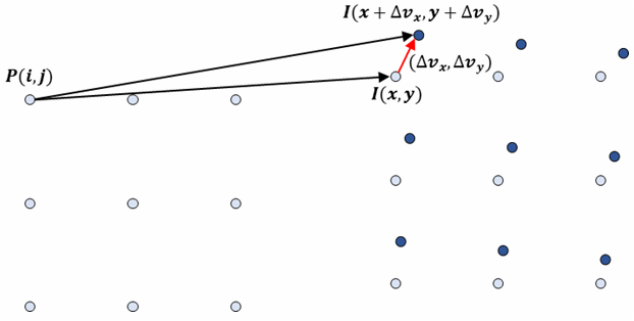


Fig. 15. Subblock AMC and sample-based AMC [22].

$$\sigma_{BDOF} = v_x\left(\frac{\partial I_0(i,j)}{\partial x}-\frac{\partial I_1(i,j)}{\partial x}\right)+v_y\left(\frac{\partial I_0(i,j)}{\partial y}-\frac{\partial I_1(i,j)}{\partial y}\right) \quad (3.3.10)$$

Finally, the bi-prediction signal of BDOF $I'_c(i,j)$ is calculated by adjusting the bi-prediction samples as follows:

$$I'_c(i,j)=\frac{1}{2}(I_0(i,j)+I_1(i,j)+\sigma_{BDOF}) \quad (3.3.11)$$

When DMVR and BDOF are applied to a CU, DMVR is performed first and followed by BDOF. If BCW, WP, CIIP, or GPM, which include the blending process, is enabled for a CU, then the BDOF is disabled. BDOF is also disabled when a CU is coded with SMVD mode.

Prediction refinement with optical flow (PROF): PROF is used to compensate for the prediction error of a subblock-based AMC with the optical flow-based sample-wise refinement. In this way, a finer granularity of AMC, which is conducted in a block-wise manner for the trade-off between prediction accuracy and complexity, can be achieved.

After the subblock-based AMC is performed, each luma prediction sample is refined by adding a difference derived based on the optical flow equation. PROF is not applied to chroma samples.

The prediction at position (i,j) in the current block $P(i,j)$ is predicted from the sample at position (x,y) in the reference picture $I(x,y)$ with the subblock MV. Let $\Delta v(i,j)$ be the difference between the sample MV computed by an affine model and the MV of the subblock to which the sample (i,j) belongs, as shown in Fig. 15. The prediction with the sample MV $I'(x,y)$ would be:

$$I'(x,y)=I(x+\Delta v_x(i,j), y+\Delta v_y(i,j))\approx I(x,y) + g_x(i,j)*\Delta v_x(i,j)+g_y(i,j)*\Delta v_y(i,j) \quad (3.3.12)$$

where $g_x(i,j)$ and $g_y(i,j)$ are the horizontal and vertical gradients of the subblock prediction, respectively, which are calculated at each sample location similar to BDOF.

In Eq. (3.3.12), the prediction refinement $\Delta I(i,j)$ is derived using the spatial gradients of each prediction sample and sample based MV offset relative to the centered subblock MV $\Delta v(i,j)$ as follows:

$$\Delta I(i,j)=g_x(i,j)*\Delta v_x(i,j)+g_y(i,j)*\Delta v_y(i,j) \quad (3.3.13)$$

The prediction refinement is added to the affine subblock prediction to form the final affine prediction as

$$I'(i,j)=I(i,j)+\Delta I(i,j) \quad (3.3.14)$$

3.4 Transform Coding

As shown in Fig. 16, H.266/VVC has adopted various transform and quantization techniques. First, after intra/inter prediction, a transform is applied to the residual signal, and the residual signals in the spatial domain are converted to the frequency domain. An integer transform based on DCT-II has been used widely in the previous video coding standards. In most video coding standards, the two dimensional (2D) transform of the residual signal is performed through two 1D transforms in the horizontal and vertical directions using separability. 1D N -point transform and inverse transform are defined in Eqs. (3.4.1) and (3.4.2), respectively.

$$F(u)=\sum_{x=0}^{N-1}p(x)v_{u,x},\ u=0,1,2,\ldots,N-1 \quad (3.4.1)$$

$$p(x)=\sum_{u=0}^{N-1}F(u)v_{u,x},\ x=0,1,2,\ldots,N-1 \quad (3.4.2)$$

where N is the transform size, $F(u)$ is the transformed coefficient, $p(x)$ is original signal, and $v_{u,x}$ is the basis element of $N\times 1$ basis vector.

The H.266/VVC transform inherits the basic framework of H.265/HEVC, such as integer transform, fixed point arithmetic operation, and intermediate data representation [23]. H.266/VVC introduced extended transform techniques to achieve better energy compaction. The new transform design of H.266/VVC is as follows.

Primary transform: The primary transform is a technique that has been used in the existing video coding, and it was named to distinguish it from the secondary transform. In H.265/HEVC, the separable transform was applied to square blocks of up to 32×32 [14]. In H.266/VVC, the maximum transform size is extended to 64×64, and non-square blocks are also supported. Furthermore, zeroing out is introduced to reduce the decoder complexity because of the increased transform size. For the 64-point transform, only the first 32 low-frequency coefficients are maintained, and the high-frequency coefficients are zeroed out. In addition to the conventional DCT-II, H.266/VVC specifies the alternative transforms, such as DST-VII and DCT-VIII. The DCT-II,

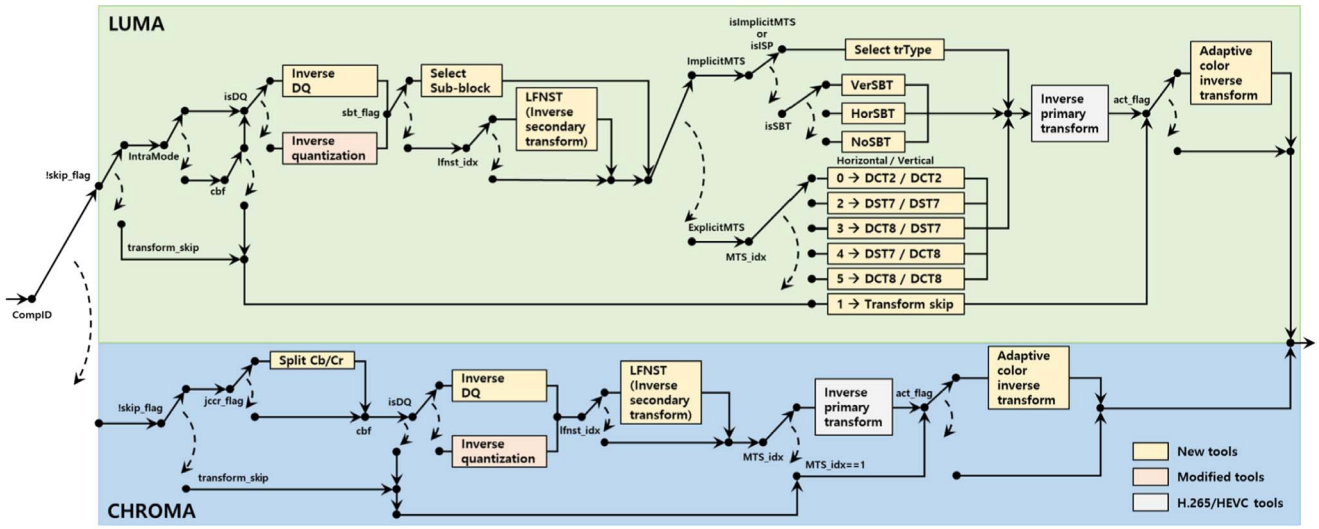


Fig. 16. Block diagram of transform and quantization in H.266/VVC decoder.

DST-VII, and DCT-VIII can be applied to the luma blocks, and only the DCT-II is used for chroma blocks. The basic function of the 1D N-point DCT-II, DST-VII, and DCT-VIII are formulated in Eqs. (3.4.3)-(3.4.5), respectively.

$$v_{u,x} = \alpha(u)\sqrt{\frac{2}{N}}\cos\left(\frac{u(2x+1)\pi}{2N}\right),$$

$$\text{where } \alpha(u) = \begin{cases} \sqrt{\frac{2}{N}}, & i = 0 \\ 1, & i \neq 0 \end{cases} \tag{3.4.3}$$

$$v_{u,x} = \sqrt{\frac{4}{2N+1}}\sin\left(\frac{(2u+1)(x+1)\pi}{2N+1}\right) \tag{3.4.4}$$

$$v_{u,x} = \sqrt{\frac{4}{2N+1}}\cos\left(\frac{(2u+1)(x+1)\pi}{4N+2}\right) \tag{3.4.5}$$

The DCT-II is applied to transform block sizes from 4×4 to 64×64, while the DST-VII and DCT-VIII are applied to transform block sizes from 4×4 to 32×32. Similar to the 64-point DCT-II, the coefficients outside the first sixteen (16) low frequency ones are zeroed out for the 32-point DST-VII and DCT-VIII.

A method that applies the transform by selecting one of the three transforms is called multiple transform selection (MTS), and there are two types of MTS, called explicit MTS and implicit MTS. The explicit MTS applies to both intra and inter coded blocks, and the selected transform combination is signaled through an index. On the other hand, implicit MTS is only available in intra coded blocks, and the transform types are derived from the coded information known to both the encoder and decoder with no index signaled. The number of multiplications and additions of multiple transforms in H.266/VVC can be found elsewhere [32].

Secondary transform: The secondary transform is a newly adopted transform tool in H.266/VVC and means an additional transform process that follows the primary

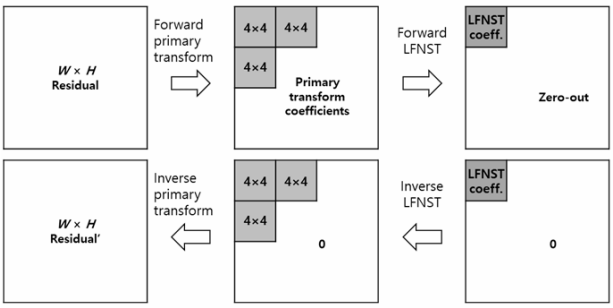


Fig. 17. Example of applying LFNST8.

transform. The low frequency non-separable transform (LFNST) is a non-separable transform that applies to the top-left low-frequency region of intra-coded blocks that use the primary transform using the DCT-II. There are two types of LFNST according to the size of the transform block. In particular, a 16×48 (row×column) kernel is applied to the top-left 8×8 region when the size of the transform block is greater or equal to 8×8. This is referred to as LFNST8. A 16×16 kernel is applied to the top-left 4×4 regions when the width or height of the transform block is 4, referred to as LFNST4. Four kernel sets are defined in LFNST according to the intra prediction modes, and two kinds of kernels exist in each kernel set. The information on whether to use LFNST and which kernel is selected is explicitly signaled through an index per CU. Similar to the zeroing out used with the primary transform, the coefficients outside of the LFNST output region are zeroed out. Fig. 17 shows an example of applying LFNST8 to a block size larger than 8×8 or equal to 8×8.

Subblock transform (SBT): SBT is a transform method that splits the inter-coded residual block and encodes only one of the two sub-partitions. There are eight modes according to the size and position of the transform block. The residual block can be divided horizontally or vertically by half or quarter size of the CU, and the coded sub-partition can be located at the left, right, top, or bottom

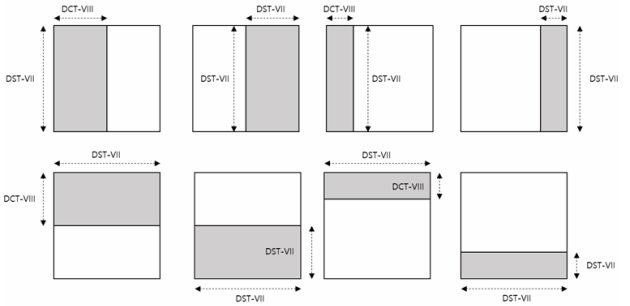


Fig. 18. SBT modes and kernel combination.

part. The transform kernel is chosen adaptively based on the location of the transform block, as shown in Fig. 18. Only DCT-II can be used for horizontal and vertical transform when the width or height of an SBT transform block exceeds 32. Fig. 18 shows eight SBT modes in H.266/VVC and selected transform combinations based on the location of the SBT transform block.

3.5 Quantization

Quantization is an irreversible operation that maps a specific range of input values to a single representative value for the input. The quantization is applied to the transform coefficients. H.265/HEVC uses scalar quantization based on uniform reconstruction quantizers (URQs). The URQs is a method that can vary the quantization rate with a single parameter called quantization step size Δ. The total number of quantization step sizes is 52, which is a real number, not an integer. A quantization parameter (QP) is an integer value from 0 to 51 and is used to prevent direct access or operation on real numbers. When QP = 4, the quantization step size equals 1, and an increase of 6 in QP means double the quantization step size. The relationship between QP and quantization step size is expressed as follows:

$$\Delta(QP) = 2^{(QP-4)/6} \tag{3.5.1}$$

In the quantization process, a quantized coefficient level (*level*) is obtained by dividing a transform coefficient C by Δ:

$$level = C / \Delta \tag{3.5.2}$$

In the de-quantization process, a reconstructed value C' can be obtained by multiplying *level* by Δ:

$$C' = level \times \Delta \tag{3.5.3}$$

The quantization scale, a value pre-multiplied by Δ and scale factor, and bitwise shift operation are used to avoid real number division.

The H.265/HEVC quantization technique is changed and added to H.266/VVC as follows [25].

Extended quantization: H.266/VVC uses URQs design from H.265/HEVC. In H.266/VVC, the maximum

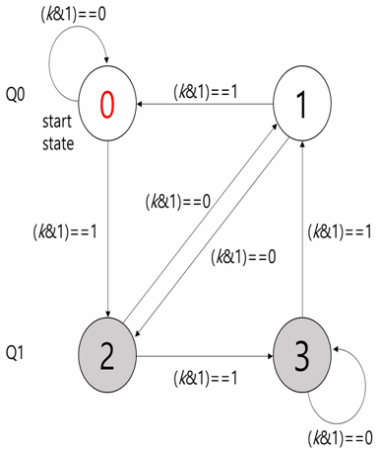


Fig. 19. State transition and quantizer selection.

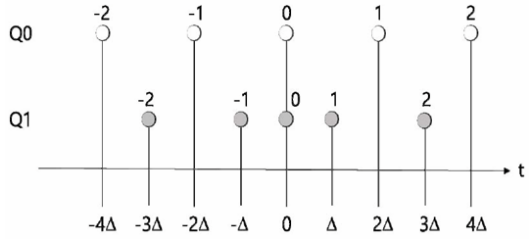


Fig. 20. Scalar quantizer Q0 and Q1.

QP was extended from 51 to 63. Adaptive frequency weighting quantization and sign data hiding (SDH) used in H.265/HEVC was also supported in H.266/VVC.

Dependent quantization (DQ): DQ defines two inverse scalar quantizers with different levels, denoted by Q0 and Q1, and enables switching between two quantizers to decode each transform coefficient. The admissible values of the current transform coefficients depend on the value of the previous transform coefficients level k. The switching between two quantizers is achieved through a state machine with four states, and the selected quantizer is not explicitly signaled. Fig. 19 shows state transition and quantizer selection; Fig. 20 shows the scalar quantizers Q0 and Q1. The initial state is set equal to 0.

Joint Coding of Chroma Residuals (JCCR): Instead of transmitting two quantized chroma residual blocks, the encoder uses JCCR mode to send one residual block using the correlation between the quantized residual signals. The decoder uses the transmitted joint residual block to generate two chroma residual blocks. The JCCR is applicable only if both chroma coded block flags (cbfs) are not zero, and H.266/VVC supports six modes according to a rotation angle. A TU-level flag indicates JCCR mode, and a selected mode is indicated by chroma cbfs and a sign of the mode.

3.6 In-loop Filtering

The block-partitioning and the quantization steps in H.266/VVC may incur undesired coding artifacts as in the previous video coding standards [3, 5]. There are four H.266/VVC in-loop filters: deblocking filter (DF), sample

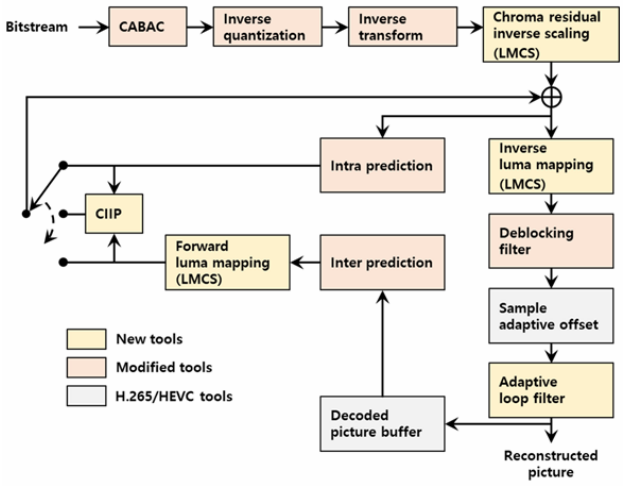


Fig. 21. Block diagram of in-loop filtering in H.266/VVC decoder [26].

adaptive offset (SAO), adaptive loop filtering (ALF), and luma mapping with chroma scaling (LMCS) [26], as shown in Fig. 21. The filters are applied during the picture reconstruction inside encoding and decoding loops in the order of inverse LMCS, DF, SAO, and ALF. The output pictures are stored in the decoded picture buffer (DPB). DF and SAO alleviate blocking artifacts and ringing artifacts as in H.265/HEVC while being adapted to new coding blocks in H.266/VVC. Compared to H.265/HEVC, ALF and LMCS are new in-loop filters adopted to H.266/VVC. ALF attempts to reduce the mean square error (MSE) between the original and the reconstructed samples by determining the filter coefficients based on a Wiener-Hopf equation [33]. The ALF provides significantly improved coding performance of H.266/VVC with approximately 4% BD-rate reductions [34], while the implementation is simplified to meet the low computational complexity. LMCS adjusts the dynamic range of pixel values in a picture to improve the objective quality of a reconstructed picture. The DF, SAO, and ALF are used in the original sample domain after the inverse LMCS because in-loop filters are originally designed to improve the subjective quality of a reconstructed picture.

H.266/VVC DF and SAO inherit the same design principles as H.265/HEVC with minor modifications to adapt to the new block partitioning. The following subsections will describe the details of ALF and LMCS.

3.6.1 Adaptive Loop Filter

ALF is based upon adaptive linear filters to restore the reconducted picture to the original picture by deriving the filter coefficients from a Wiener-Hopf equation. The output samples of SAO are used as input samples, as shown in Fig. 21. When the filter coefficients are transmitted to a decoder, the derivation process of filter coefficients can be conducted using RDO to improve the coding performance. Despite the improved coding performance, the derivation process requires heavy computational loads for a practical real-time decoder of a consumer device. The filtering mechanisms are simplified

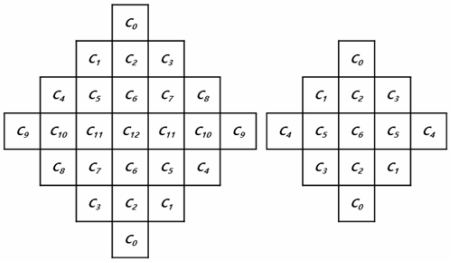


Fig. 22. Filter shapes and sizes in VVC ALF: 7×7 and 5×5 symmetric diamond-shaped filters for luma and chroma samples, respectively.

during the H.266/VVC standardizations while maintaining the coding performance.

The ALF uses two 2D finite impulse response (FIR) filters with a 7×7 diamond shape and a 5×5 diamond shape applied to the luma and chroma samples, respectively [35]. Fig. 22 exhibits the two filter shapes to correct the center samples. The filter shapes and sizes are determined from extensive experiments with various resolutions of test videos to consider the trade-off between the coding efficiency and computational complexity [36, 40]. The ALF uses symmetric FIR filters, in which the numbers of coefficients are 13 and 7 for the 7×7 and 5×5 filter shapes, respectively, which reduces the computational complexity [38, 39]. A line buffer reduction is also applied to reduce the storage requirements for ALF [40].

The spatially neighboring samples to the center points are used for deriving the corresponding filter coefficients c_i in Fig. 22. A filtered sample $\tilde{I}(x,y)$ in the current position is corrected from the reconstructed sample $I(x,y)$ using the weighted linear combinations of c_i with a 7-bit fractional precision and the spatially neighboring samples. Once c_i is derived from the Wiener-Hopf equation, the filtering is defined as

$$\tilde{I}(x,y) = I(x,y) + \left[\left(\sum_{i=0}^{N-2} c_i r_i + 64\right) \gg 7\right] \qquad (3.6.1)$$

where N is 13 and 7 for the luma and chroma samples, respectively. r_i refers to a difference between the current pixel and a neighboring sample specified in Fig. 22. Specifically, r_i is computed as follows:

$$r_i = \min\left(b_i, \max\left(-b_i, I(x+x_i, y+y_i) - I(x,y)\right)\right) + \min\left(b_i, \max\left(-b_i, I(x-x_i, y-y_i) - I(x,y)\right)\right) \qquad (3.6.2)$$

where b_i refers to a clipping parameter determined by a clipping index d_i and a sample bit depth BD. The clipping parameter b_i is set to 2^{BD} when $d_i = 0$. Otherwise (d_i = 1, 2, and 3), it is $2^{BD-1-2d_i}$. The clipping index is transmitted to a decoder.

The ALF maintains a set of up to 25 filter coefficients for luma samples, which can be applied adaptively to each 4×4 subblock. A 4×4 subblock is categorized into one of 25 classes. The classification is conducted to obtain directions and activity using local gradients with Laplacian filters. Specifically, the classification index is derived from a combination of five directional properties, including a texture, strong and weak horizontal/vertical, strong and weak diagonal, and five activity properties of a subblock. As a result, a different filter can be assigned to each class. Furthermore, geometric transform, such as 90-degree rotation, diagonal, or vertical flip, can be applied to the filter coefficients before the filtering. Various directionality can be considered using the geometric transform, and the ALF handles more diverse block characteristics with fewer filter coefficients using this method [41].

In addition to a subblock adaptation, for a coding tree block (CTB)-level adaptation, a set of filter coefficients is obtained online using the current slice or the previous slices or 16 offline trained filter sets. For chroma samples, H.266/VVC ALF uses only the CTB-level filter adaptation using up to eight filters [42]. In H.266/VVC, the adaptation parameter set (APS) [19] is used to carry the ALF filter parameters, which include up to 25 and eight sets of filter coefficients for luma and chroma components, respectively, and clipping indices are signaled. When the same ALF coefficients are used for different slices, only the ID of a reference APS can be signaled instead of a redundant transmission. Furthermore, the ALF is controlled using on/off flags signaled in sequence, picture, slice, and CTB levels. Chroma ALF is enabled only when the luma ALF is enabled at the corresponding level.

H.266/VVC supports a wider variety of video applications with HDR and WCG, for which the in-loop filters attempt to improve the visual quality of both luma and chroma samples. A cross-component adaptive loop filtering (CC-ALF) [43] corrects the chroma samples in parallel with the ALF, using the correlation between the current chroma samples and the luma samples in the corresponding positions. CC-ALF applies a linear filtering operation to produce the correction of chroma samples from the luma samples as input. The CC-ALF uses diamond-shaped FIR filters without symmetric constraints.

3.6.2 Luma Mapping with Chroma Scaling

LMCS has been adopted to H.266/VVC to improve the coding efficiency by processing the dynamic ranges of input samples rather than improving visual quality directly [44]. The LMCS has been originally proposed to improve the coding efficiency of HDR and WCG PQ video contents in which most input video samples tend to be distributed within a relatively narrow range compared with the SDR video contents [45]. LMCS supports both HDR and SDR video content in H.266/VVC.

The LMCS consists of a luma mapping (LM) module and a chroma scaling (CS) module. The LM module maps luma code values from the original sample domain to an LMCS domain using a forward luma mapping (*FwdMap*), or vice versa using an inverse luma mapping (*InvMap*). In

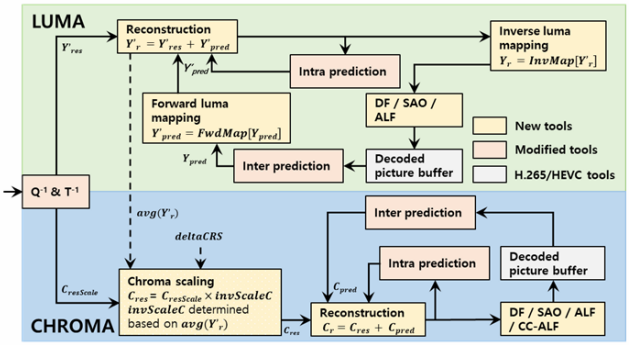


Fig. 23. LMCS in H.266/VVC decoder [26].

the CS module, a luma-dependent chroma residual scaling is applied to balance the impact of luma remapping [46].

Fig. 23 depicts the LM and CS process in H.266/VVC decoder. The shadowed regions, including inverse quantitation and transform, an intra prediction, and the luma reconstruction of an intra prediction, represent decoding processes under an LMCS domain [45]. Accordingly, an inter prediction and intra prediction signals go through different procedures with LMCS. For example, a residual signal and an inter prediction signal are obtained in an LMCS domain and the original domain, respectively. *FwdMap* is conducted to convert a domain of a reference signal in a DPB for motion compensation, and the reconstruction signals are stored in a DPB after *InvMap* .

FwdMap is determined using an adaptive piecewise linear model derived from related syntax elements of LMCS APS according to a dynamic range of input video samples. Specifically, the original code values are sampled uniformly into 16 pieces to calculate *OrgCW* . For each piece i , the number of mapped code values is defined as $binCW[i]$. To define *FwdMap* , the slope $scaleY[i]$ is calculated as

$$scaleY[i] = \frac{binCW[i]}{OrgCW} \tag{3.6.3}$$

and $invScaleY[i]$ as the slopes of *InvMap* is calculated with the inverse of $scaleY[i]$. In a decoder, $scaleY[i]$ and $invScaleY[i]$ can be derived because the difference between *OrgCW* and $binCW[i]$ is signaled in the LMCS adaptive parameter set (APS).

For CS, forward and inverse scaling are applied to the chroma residue with a factor of *ScaleC* and *invScaleC* , respectively. In H.266/VVC, *invScaleC* is defined as

$$invScaleC[i] = \frac{OrgCW}{binCW[i] + deltaCRS} \tag{3.6.4}$$

where *deltaCRS* is a chroma scaling offset value. In this manner, as shown in Fig. 23, a chroma residue-scaled

value is produced from a decoding process, and a chroma residue value is calculated by multiplying *invScaleC* [47].

3.6.3 Towards Deep Learning In-loop Filter

Current in-loop filters are designed to perform a pixel-level restoration of a reconstructed image. Deep convolutional neural network (CNN) has attracted considerable attention from video coding experts because many data-driven approaches for image denoising have been actively studied in image processing and computer vision research areas [48]. During H.266/VVC standardization, the benefit of deep learning techniques in video compression has been discussed in AhG9 [48, 49], and various CNN-based in-loop filters have been tested extensively in core experiments [50] to investigate the coding efficiency and computational complexity.

Although the CNN-based in-loop filter has not been adopted to the H.266/VVC specification, it envisioned a future research and development direction of in-loop filters in hybrid video coding standards. Currently, video coding experts continue verifying the effectiveness of neural network video coding (NNVC) [51].

3.7 Screen Contents Coding Tools

A screen content coding (SCC) tool can efficiently encode computer-generated video that exhibits different signal characteristics from the usual video captured by a camera. Screen content video mainly contains characters, lines, graphs, and patterns. Thus, it is characterized by sharp edges, uniformly flat areas, repeating patterns, and highly saturated colors, most of which are rarely found in camera-captured images/videos. To address its rather unique characteristics, new coding tools have been added to H.265/HEVC, i.e., HEVC range extension (REext) and SCC extensions [52]. Through much improvement and refinement, the tools that have been adopted to H.266/VVC are as follows: Transform skip residual coding (TSRC), block-based differential pulse-coded modulation (BDPCM), intra block copy (IBC), adaptive color transform (ACT), and palette mode coding [53]. TSRC is integrated with the transform skip mode (TSM), which has existed since H.265/HEVC. IBC, ACT, and the palette mode are inherited from the H.265/HEVC SCC extensions. This section describes the five main screen content coding tools in H.266/VVC.

3.7.1 Transform Skip Residual Coding

Transform skip residual coding (TSRC) is a CABAC entropy coding scheme designed especially for transform skip residual blocks. In the H.265/HEVC REext extension, by producing a dedicated context model for flags representing absolute values greater than zero and 180-degree rotating intra predicted transform skip residuals, statistical difference between regular residual coding (RRC) and TSRC has already been considered, but only partially. TSRC, which is newly introduced to H.266/VVC, directly addresses the difference by employing the following three main features. First, instead of transmitting the last significant scan position, it encodes the

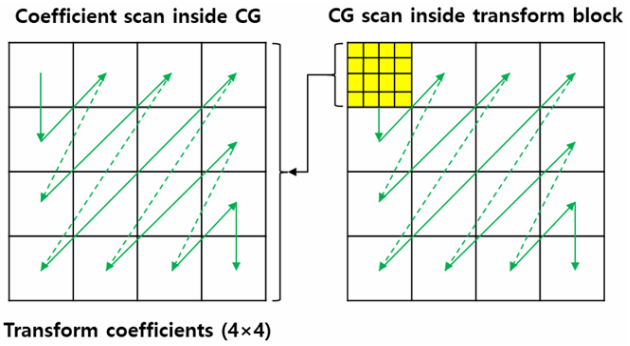


Fig. 24. Coefficient group scan for TSRC in H.266/VVC.

quantization indices of all scan positions of a transform block. As shown in Fig. 24, the scanning direction is from top-left to bottom-right, which is a reversed version of an RRC.

Second, even when the global distribution is almost uniform, the non-stationarity of the symbol makes it possible to code the symbol indicators more efficiently using the context model. Last, the binarization of absolute level values is changed by coding more context-coded "greater than x" flags and modifying the Rice parameter derivation for the Golomb-Rice code suffix, resulting in a higher cutoff for the unary binarization prefix.

3.7.2 Block Differential Pulse Coded Modulation

Block differential pulse coded modulation (BDPCM), especially useful for screen content, is one of the intra prediction modes in H.266/VVC. It aims for better decorrelation of the intra-predicted residual of the screen content by replacing the usual transform of DCT or DST. Whether to do BDPCM or not is signaled by a flag at the CU level. In case BDPCM is used, an additional flag further signals the direction of block prediction in BDPCM. The BDPCM predictor is generated through either horizontal or vertical prediction using unfiltered reference samples. The residual values are quantized, and the difference signal $\tilde{r}_{i,j}$ is calculated using Eq. (3.7.1) for vertical BDPCM and Eq. (3.7.2) for horizontal BDPCM.

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0 \\ Q(r_{i,j}) - Q\left(r_{(i-1),j}\right), & 1 \le i \le (H-1) \end{cases} \quad (3.7.1)$$

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & j = 0 \\ Q(r_{i,j}) - Q\left(r_{i,(j-1)}\right), & 1 \le j \le (W-1) \end{cases} \quad (3.7.2)$$

Here, $r_{i,j}$ denotes the intra-predicted residual signal of a block at a position $(0 \le i \le H-1,\ 0 \le j \le W-1)$ inside the block. The block has a size $H \times W$, and $Q(\cdot)$ denotes the quantization operation. The difference signal is transmitted to the decoder using TSRC. In the decoding process, the quantized residual $Q(r_{i,j})$ is reconstructed using Eq. (3.7.3) for vertical BDPCM and Eq. (3.7.4) for

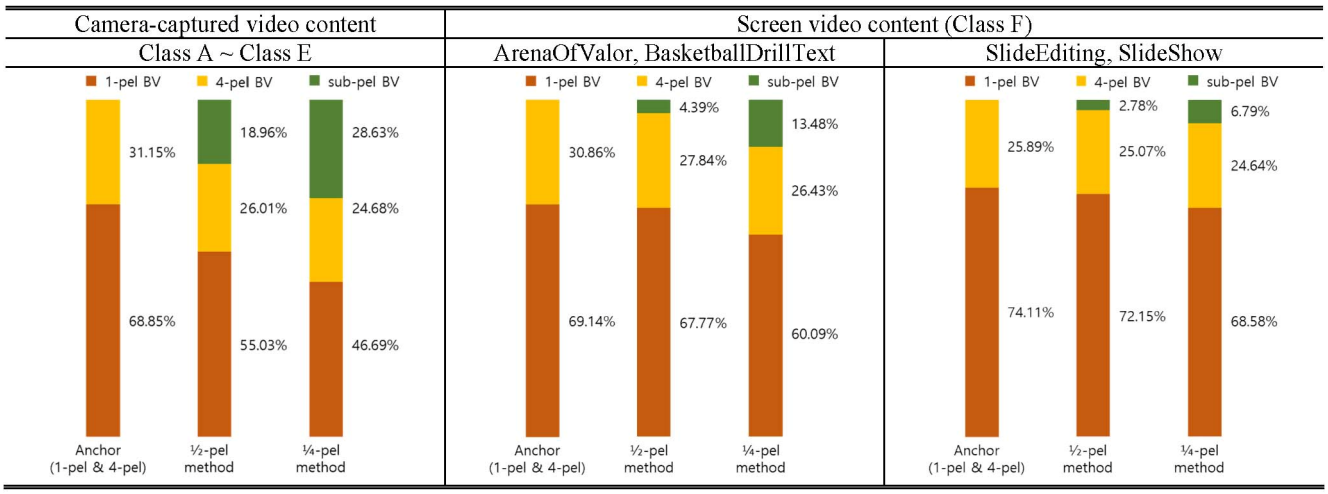


Fig. 25. Average ratio of BV resolution in camera-captured and screen video contents [54].

horizontal BDPCM.

$$Q\left(r_{i,j}\right)=\sum_{k=0}^{i}\tilde{r}_{k,j} \tag{3.7.3}$$

$$Q\left(r_{i,j}\right)=\sum_{k=0}^{j}\tilde{r}_{i,k} \tag{3.7.4}$$

The dequantized residual $Q^{-1}\left(Q\left(r_{i,j}\right)\right)$ is added to the intra prediction to generate the reconstructed sample.

3.7.3 Intra Block Copy

Intra block copy (IBC) finds a reference block similar to a current block inside a designated area of the current frame. The reference block works as an intra-predictor of IBC, and its location is represented by a block vector (BV), which specifies its displacement from the current block. The difference in BV from its block vector predictor (BVP), noted as the block vector difference (BVD), is encoded using either the merge mode or the AMVP mode. The merge mode is the most efficient coding tools for the BV vector because it only sends the merge index with its BVD treated as zero (thus, not encoded). The merge index indicates one vector for BVP in the list of six merge candidates. The merge candidate list includes the following until it has six members in the order of the BVs of a spatially adjacent block (bottom-left and top-right), history-based BVs, and zero vectors. The AMVP mode transmits the AMVP flag and BVD, where the AMVP flag indicates one selected BVP in the AMVP candidate list of size two.

Because the IBC is developed as an intra prediction coding tool specialized for screen content, its encoder only searches the reference blocks at integer positions. Noting that the screen content is generated pixel-by-pixel by a computer, no consideration is given to representing the BV vector in the sub-pel resolution. The AMVR allows the BVD to be encoded at a higher resolution, such as in a 1-pel or 4-pel resolution. On the other hand, it is interesting to investigate the effectiveness of sub-pel BV resolution by further estimating the BV into the half-pel and quarter-pel resolution. In this context, the experiment results for screen content video [54] confirm that compared to camera-generated video, both half-pel and quarter-pel BVs are chosen less often in actual encoding, even if they can be used, but it is not absolute. Therefore, the screen content is rendered more by sophisticated rendering techniques, which are often used in recent screen content, as shown in Fig. 25. That is, those screen contents with many characters, lines, and graphics like ‘SlideEditing’ (this type of screen content can be referred to as legacy screen content) have chosen much fewer sub-pel BVs and permit additional BV resolutions in encoding, resulting in coding loss [53]. On the other hand, more natural-looking computer-generated video by sophisticated rendering processes is seen to choose sub-pel BVs, not trivially, as shown in Fig. 25 [54].

3.7.4 Adaptive Color Transform

Adaptive color transform (ACT) is a particularly effective technique for video sequences expressed in the RGB color space because it can effectively reduce the correlation among the three color components in 4:4:4 chroma format. The ACT technique in H.266/VVC has existed since H.265/HEVC-SCC. It selectively transforms the residual signal in the input color space (RGB) into the YCgCo-R luma-chroma color representation according to a CU-level flag. The maximum ACT size cannot exceed 32×32 samples to ease the cache requirement of temporarily storing all three transform blocks. The YCgCo-R transform is fully reversible and can be applied to lossless coding.

3.7.5 Palette Mode

Palette mode, one of the intra prediction modes, supports all chroma formats. Considering the relatively insignificant coding gain in small blocks and the complexity of the palette mode, it is not applied unless a CU has more samples than 16. Each sample in a CU coded in palette mode needs to signal its palette index with a

Table 4. Ratios of intra and inter prediction.

	Intra prediction	Inter prediction
LB	2.77%	97.23%
LP	3.07%	96.93%
RA	13.35%	86.65%

representative color set called a palette. Otherwise, an escape symbol is signaled. If a sample is coded using escape symbols, its quantized component values are signaled. The palette is defined separately for the luma (Y component) and chroma (Cb and Cr components) for slices in a dual tree. In this case, the entries in the luma palette hold just Y values and the chroma palette entries include Cb and Cr values. For the single tree slices, a palette entry contains Y, Cb, and Cr values, and the palette is applied to Y, Cb, and Cr components jointly. For the single tree, the maximum palette predictor size is 63, and the maximum palette table size for a CU is 31. For the dual tree, the maximum predictor size and table size are 31 and 15, respectively.

4. Performance Evaluation of H.266/VVC

The various coding techniques of the H.266/VVC previously described are more complex than those included in the H.265/HEVC standard. Efficient encoding search algorithms that can reduce the encoding time while maintaining compression efficiency are needed to implement a real-time H.266/VVC encoder, particularly an early termination strategy developed that considers the coding efficiency and tool selection ratio for each technique. Therefore, this section evaluated the performed proportion and coding efficiency of coding techniques for each module of H.266/VVC. First, an analysis of the selection ratio of each coding technique is performed using the bitstreams generated by the VTM12.0 encoder in JVET common test condition (CTC) [55]. The selection ratio for each tool is normalized to 4×4 for each CU or TU to indicate the ratio of the area where each coding tool is selected in the entire sequence. In addition, the experiments are conducted using the VTM12.0 encoder and decoder in JVET CTC to evaluate the coding performance according to the block partitioning structure, CTU and TU sizes, and a group of pictures (GOP) size. In addition, all the experiments are conducted in a cluster environment consisting of Intel® i7-10, 10700K 3.8Hz processors, and 64G RAM. The class F sequences, which use the screen content tools, are excluded from the selection ratio analysis to analyze commonly used techniques for the different content types.

4.1 Prediction Modes

This study first examines the overall selection ratio of intra and inter prediction before analyzing each prediction tool according to encoding configurations. Table 6 shows the selection ratios of intra and inter prediction in all classes of JVET CTC. In the LB and LP configurations, where all frames except the first frame consist of B-frames and P-frames, the intra predicted region ratio is 2.77% and 3.07%, respectively.

On the RA configuration, where I-frames are inserted for each random access point (RAP), the intra predicted region ratio is 13.35%. In particular, the proportion of intra predicted regions in class A1 is about 27% on the RA configuration. This ratio is higher than the other sequences because the class A1 sequences consists of many complex textures. The other sequences show that the proportion of intra predicted regions is less than 10%. Accordingly, when designing the H.266/VVC encoder, it is expected that the encoding complexity can be reduced by adjusting the ratio of rate-distortion optimization (RDO) to the early termination conditions of the intra and inter prediction based on the target video for applications.

4.2 Intra Prediction Tools

Tables 7 and 8 show the selection ratio of intra prediction tools for the luma and chroma components, respectively. On the AI and RA configurations, the selected intra prediction tool per region is similar for each class. In particular, the planar, angular prediction, and MIP modes are selected as the highest in the luma component. In the case of angular prediction mode, more than 30% is selected in the entire intra-predicted region. Specifically, the performed proportion of horizontal mode among the angular predicted regions is approximately 30% and 60% on AI and RA configurations, respectively. For MIP mode, the matrices in each case, as defined in Table 3, are selected almost equally for each sequence, with an average of 6.25%, 12.5%, and 16.67% on both configurations. In the MRL tool, the selection ratio of the farther away non-adjacent reference lines is double that on all configurations. In addition, the error correction through PDPC is more than 70% of the entire intra-predicted region, and the signal of the prediction mode through the MPM list is shown to be 70% or more. In the chroma component, the performed proportion of derived mode (DM) and CCLM mode is the highest, and the selection ratio of non-angular (DC, planar) and angular prediction mode is similar except for Class A1 in the case of DM mode. In addition, the selection ratio of CCLM and PDPC is more than 40% and over 85%, respectively.

In addition, the MIP mode is turned off on the LB and LP configurations according to JVET CTC. As a result, the selection ratios of planar and angular prediction modes show larger than that of AI and RA configurations. The angular prediction modes are selected for more than 40% of the entire intra-predicted region. Moreover, the error correction through PDPC is more than 70% of the entire intra-predicted region, and the case of signaling the prediction mode based on the MPM list is more than 68%. In the chroma components, the DM and CCLM modes had the highest performed proportion, and the ratio of both tools is more than 80%, indicating that they are selected as the prediction mode for most regions. In addition, the PDPC is applied at higher rates than 70% on the LB and LP configurations, respectively, as on the AI and RA

Table 7. Ratios of intra prediction tools for the luma component.

	Prediction mode															PDPC	MRL		MPM
	DC	Planar	Ang.		MIP			ISP											
			Reg.	Wide	Case1	Case2	Case3	Hor. (Pred./Transf.)				Ver. (Pred./Transf.)					Line1	Line2	
								1/2	1/4	2/4	4/4	1/2	1/4	2/4	4/4				
AI	5.02%	23.72%	29.18%	1.45%	0.10%	2.38%	23.13%	0.48%	7.02%	3.24%	1.33%	0.38%	0.66%	0.93%	0.97%	77.82%	1.95%	4.46%	73.72%
LB	7.03%	35.22%	44.28%	2.98%	0.00%	0.00%	0.00%	0.24%	0.69%	2.39%	1.44%	0.21%	0.98%	2.74%	1.80%	70.54%	1.42%	1.89%	68.13%
LP	7.13%	34.51%	44.88%	3.07%	0.00%	0.00%	0.00%	0.23%	0.68%	2.42%	1.40%	0.19%	0.99%	2.77%	1.73%	70.16%	1.48%	1.94%	67.53%
RA	2.94%	19.24%	36.87%	1.41%	0.03%	1.45%	17.64%	0.29%	10.70%	3.99%	3.07%	0.23%	0.54%	0.71%	0.89%	70.62%	6.25%	13.09%	79.32%

Table 8. Ratios of intra prediction tools for the chroma component.

	Prediction mode										PDPC
	DC	Planar	Hor.	Ver.	DM			CCLM			
					Non-Ang.	Ang.		T	L	LT	
						Reg.	Wide				
AI	4.42%	3.26%	3.41%	3.34%	16.36%	23.65%	0.65%	9.94%	9.69%	25.28%	86.34%
LB	1.28%	1.74%	0.90%	1.48%	32.18%	33.22%	2.23%	5.65%	4.95%	16.38%	70.61%
LP	1.24%	1.68%	0.87%	1.43%	32.03%	33.91%	2.32%	5.59%	4.84%	16.09%	70.00%
RA	1.68%	1.68%	5.68%	1.14%	12.36%	39.14%	1.29%	8.05%	7.42%	21.57%	68.60%

Table 9. Ratios of the reference types and motion information coding modes.

	Reference type		Uni-prediction			Bi-prediction		
	Uni	Bi	Merge	Merge/skip	AMVP	Merge	Merge/skip	AMVP
LB	25.22%	74.78%	6.67%	12.82%	5.73%	18.16%	48.42%	8.20%
LP	100%	0.00%	25.91%	64.01%	10.09%	0.00%	0.00%	0.00%
RA	19.47%	80.53%	4.66%	9.10%	5.71%	13.99%	56.98%	9.56%

Table 10. Ratios of the inter prediction tools in the uni-directional prediction.

Uni	Merge						Merge/skip					AMVP		
	Regular	MMVD	CIIP	SbTMVP	Aff. 4	Aff. 6	Regular	MMVD	SbTMVP	Aff. 4	Aff. 6	Regular	Aff. 4	Aff. 6
LB	10.44%	6.53%	2.85%	0.00%	1.53%	5.10%	30.07%	7.84%	0.00%	3.68%	9.25%	10.46%	5.58%	6.68%
LP	11.84%	4.41%	2.93%	2.58%	1.04%	3.11%	39.75%	5.61%	11.28%	2.35%	5.02%	4.44%	2.48%	3.17%
RA	11.15%	5.50%	2.54%	0.86%	1.06%	2.83%	28.22%	4.20%	8.78%	1.94%	3.59%	16.14%	4.55%	8.63%

Table 11. Ratios of the inter prediction tools in the bi-directional prediction.

Bi	Merge							Merge/skip						AMVP			
	Regular	MMVD	GPM	CIIP	SbTMVP	Aff. 4	Aff. 6	Regular	MMVD	GPM	SbTMVP	Aff. 4	Aff. 6	Regular	Aff. 4	Aff. 6	SMVD
LB	10.46%	3.73%	3.09%	1.63%	3.02%	0.77%	1.58%	38.63%	5.07%	3.33%	13.69%	1.62%	2.41%	6.07%	2.50%	2.40%	0.00%
RA	9.93%	3.50%	0.85%	0.62%	0.65%	0.54%	1.28%	56.32%	3.99%	1.03%	7.03%	0.75%	1.35%	3.45%	1.26%	2.10%	5.07%

configurations. For all configurations in JVET CTC, setting early termination conditions considering the ratio will be reduced the encoding complexity because the MPM list based prediction for the luma component and CCLM modes for the chroma component, and the correction through PDPC for all components apply to most intra-predicted regions.

4.3 Inter Prediction Tools

In order to analyze the selection ratio of each detailed inter prediction tool according to the reference type and the motion coding methods (merge, merge/skip, and AMVP) in the inter-predicted regions, the proportion of the uni-/bi-directional predicted regions, and the selection ratio of the motion information coding methods are first examined according to the reference type, as shown in Table 9. In the LB and RA configurations, where both uni-directional and bi-directional prediction is available, 25.22% and 19.47% of the uni-directional prediction modes are applied, respectively. In addition, the merge/skip mode is most selected for motion coding in all conditions. Tables 10 and 11 present the proportion of each inter prediction tool according to motion coding methods.

Table 12. Ratios of BCW weights.

		BCW weight				
		-2	3	4	5	10
LB	Merge	0.05%	2.81%	85.96%	9.47%	1.71%
	Merge/skip	0.04%	1.53%	89.17%	7.39%	1.88%
	AMVP	0.29%	10.15%	57.43%	26.95%	5.18%
RA	Merge	0.02%	10.53%	77.91%	11.49%	0.05%
	Merge/skip	0.00%	4.20%	91.31%	4.47%	0.02%
	AMVP	0.07%	23.45%	51.31%	25.10%	0.08%

Table 13. Ratios of BDOF and DMVR.

	BDOF	DMVR
Merge	36.15%	40.02%
Merge/skip	10.56%	72.11%
AMVP	68.69%	0.00%

Table 14. Ratios of PROF for the uni-directional AMC-predicted CU.

Uni	Merge				Merge/skip				AMVP			
	Affine 4-parameter		Affine 6-parameter		Affine 4-parameter		Affine 6-parameter		Affine 4-parameter		Affine 6-parameter	
	L0	L1	L0	L1	L0	L1	L0	L1	L0	L1	L0	L1
LB	94.42%	85.64%	98.21%	93.98%	96.02%	87.05%	98.23%	90.05%	97.72%	0.00%	99.97%	0.00%
LP	80.95%	0.00%	94.13%	0.00%	80.16%	0.00%	94.02%	0.00%	96.09%	0.00%	99.34%	0.00%
RA	41.21%	24.85%	67.64%	50.21%	23.04%	13.53%	41.54%	30.54%	28.23%	18.47%	53.79%	40.86%

Table 15. Ratios of PROF for the bi-directional AMC-predicted CU.

Bi	Merge						Merge/skip						AMVP					
	Affine 4-parameter			Affine 6-parameter			Affine 4-parameter			Affine 6-parameter			Affine 4-parameter			Affine 6-parameter		
	Only L0	Only L1	Both	Only L0	Only L1	Both	Only L0	Only L1	Both	Only L0	Only L1	Both	Only L0	Only L1	Both	Only L0	Only L1	Both
LB	2.75%	22.68%	56.64%	0.80%	8.49%	88.19%	2.90%	24.90%	51.51%	1.02%	11.93%	83.46%	3.33%	52.25%	40.07%	0.56%	41.56%	57.72%
RA	2.88%	3.12%	15.85%	4.84%	3.95%	44.80%	2.04%	1.65%	9.50%	4.01%	2.91%	34.12%	4.15%	3.77%	14.73%	5.65%	3.59%	46.52%

The most often selected inter prediction tool is the regular mode selected up to 56.32%, which performed the whole block-based motion compensation through the coded motion without the refinement tool. Among the inter prediction tools, AMC (e.g., Aff.4 and Aff.6) and GPM modes showed the highest coding efficiency [34], but they have a lower selection ratio under all conditions. This suggests that a higher coding efficiency compared to the selection proportion is obtained by performing a more accurate prediction through the reference picture generated by applying AMC and GPM modes.

Table 12 shows the signaled weight (w) ratio when the bi-directional prediction is performed using Eq. (3.2.2) using BCW. Both P_{L0} and P_{L1} had the highest selection ratio of 91.31% when weighted using the same weight, and the lowest selection ratio when the weighted prediction using -2 and 10, where the weights of P_{L0} and P_{L1} differ the most. On the other hand, the VTM12.0 encoder performs RDO in the ascending order of each index value without considering these selection ratios. Therefore, if the performed proportion is considered in future H.266/VVC encoder designs, the encoding complexity will be improved while maintaining the coding performance.

Tables 13-15 show the selection ratio of the following: the BDOF, a tool for performing motion refinement on the decoder side based on optical flow; the PROF, which performs sample-wise refinement within an AMC block based on optical flow; the DMVR, which performs motion refinement through bilateral matching. BDOF and DMVR can only be applied to the RA configuration. The proportion of motion refinement performed through the BDOF tool is 10.56% in merge/skip mode and 68.69% in AMVP, as shown in Table 13, showing a significant difference in the ratio depending on the motion coding methods. The DMVR tool showed a high selection ratio of 72.11% in merge/skip mode, which skipped the transform through the skip flag. This result is showed that even if DMVR is applied by generating more accurate predictors through motion refinement, the CU can be reconstructed to be close to the original signal without the signaled MVD. Tables 14 and 15 lists the proportions of PROF for AMC-predicted CUs according to reference type. For uni-

Table 16. Ratios of the AMVR index for non AMC-predicted CU.

	AMVR (for non-affine)			
	1/4-pel	1-pel	4-pel	Half-pel
LB	76.50%	13.25%	3.53%	6.72%
LP	69.09%	17.69%	3.03%	10.19%
RA	49.33%	31.31%	11.27%	8.08%

Table 17. Ratios of the AMVR index for AMC-predicted CU.

	AMVR (for affine)		
	1/4-pel	1/16-pel	1-pel
RA	59.85%	8.14%	32.01%

Table 18. Ratios of the transform tools of the intra-predicted CUs for the luma component.

	Transform skip	MTS		Transform kernel						LFNST
		Implicit	Explicit	Horizontal			Vertical			
				DCT2	DCT8	DST7	DCT2	DCT8	DST7	
AI	24.02%	33.83%	66.17%	16.18%	0.0012%	83.82%	21.28%	42.27%	36.45%	18.31%
LB	17.50%	14.23%	85.77%	2.85%	4.66%	92.50%	5.03%	29.77%	65.20%	0.00%
LP	17.99%	14.33%	85.67%	2.87%	4.69%	92.43%	5.00%	29.59%	65.40%	0.00%
RA	27.13%	44.08%	55.92%	31.06%	0.0011%	68.94%	31.89%	37.61%	30.50%	15.79%

Table 19. Ratios of the transform tools of the inter-predicted CUs for the luma component.

	Transform skip	SBT									Transform kernel					
		SBT on	Half				Quad				Horizontal			Vertical		
			Vertical		Horizontal		Vertical		Horizontal							
			Left	Right	Top	Down	Left	Right	Top	Down	DCT2	DCT8	DST7	DCT2	DCT8	DST7
LB	70.96%	3.22%	29.12%	13.67%	25.45%	11.96%	7.18%	3.27%	6.55%	2.82%	94.04%	1.64%	4.32%	94.04%	1.44%	4.51%
LP	71.47%	2.99%	29.13%	13.58%	25.74%	11.91%	7.09%	3.24%	6.56%	2.76%	94.31%	1.56%	4.13%	94.31%	1.39%	4.30%
RA	80.64%	1.73%	28.12%	12.62%	26.97%	11.77%	7.13%	3.23%	7.12%	3.04%	95.41%	1.28%	3.31%	95.41%	1.16%	3.43%

directional predictions in the LB and LP configurations, most AMC-predicted CUs have performed PROF. In particular, up to 99.97% of the AMC-predicted CUs have performed PROF in AMVP mode. Furthermore, PROF is performed mainly on AMC-prediction CUs in the L0 and L1 directions rather than L0 or L1 for bi-directional predictions in the LB and LP configurations.

Tables 16 and 17 show the selection ratio of precision of MVDs signaled to the index through AMVR in the AMVP mode. According to JVET CTC, sps_affine_amvr_enabled_flag is set to 0 on LB and LP configurations. Therefore, the precision of MVD is fixed to 1/4-pel for the AMC-predicted CU. For the RA configuration, 1/4-pel is the precision most frequently selected for AMC-predicted CU, with 59.85%, followed by 1-pel with 32.01%. Moreover, for all configurations, the precisions are selected in the order of 1-pel, half-pel, and 4-pel on the LP and LB configurations, and 1-pel, 4-pel, and half-pel are selected in the RA configuration.

4.4 Transform Tools

Tables 18-21 show the selection ratio of the transform tools in luma and chroma components for each region, respectively. Specifically, Tables 18 and 19 show the performed proportion of the transform tools for the residual signal of the intra-/inter-predicted CUs of the luma component, and Tables 20 and 21 show the selection

Table 20. Ratios of transform tools of the intra-predicted CUs for the chroma components.

	Transform skip	LFNST
AI	58.96%	20.55%
LB	87.46%	0.00%
LP	87.85%	0.00%
RA	67.28%	2.42%

Table 21. Ratios of transform tools of the inter-predicted CUs for the chroma components.

	Transform skip
LB	96.91%
LP	96.96%
RA	91.07%

Table 22. Ratios of SAO.

	SAO (Luma)								SAO (Chroma)							
	New	Merge	Off	EO (0°)	EO (90°)	EO (135°)	EO (45°)	BO	New	Merge	Off	EO (0°)	EO (90°)	EO (135°)	EO (45°)	BO
AI	16.68%	80.08%	3.24%	66.63%	19.80%	3.48%	4.09%	2.77%	7.56%	80.08%	12.36%	64.08%	18.04%	1.45%	1.49%	2.58%
LB	6.57%	22.04%	71.39%	18.26%	7.13%	1.08%	1.06%	1.08%	0.96%	6.66%	92.39%	5.14%	1.69%	0.21%	0.23%	0.35%
LP	8.02%	26.04%	65.94%	21.68%	8.60%	1.40%	1.51%	0.86%	1.10%	7.34%	91.56%	5.61%	1.96%	0.25%	0.27%	0.36%
RA	3.78%	26.65%	69.57%	23.37%	4.81%	0.62%	0.70%	0.93%	3.34%	23.46%	73.20%	20.48%	4.29%	0.54%	0.62%	0.87%

Table 23. Ratios of ALF and CCALF for the chroma components.

		ALF (Chroma)								CC-ALF				
		Off	Filter index							Off	Filter index			
			1	2	3	4	5	6	7		1	2	3	4
AI	Cb	59.30%	9.59%	5.98%	4.39%	3.65%	4.15%	5.02%	7.93%	38.26%	1.31%	0.98%	2.05%	57.41%
	Cr	59.37%	11.87%	6.66%	4.75%	4.31%	3.86%	3.90%	5.28%	36.31%	0.00%	0.02%	1.09%	62.58%
LB	Cb	89.67%	4.84%	2.16%	1.20%	0.84%	0.56%	0.42%	0.30%	50.62%	1.51%	1.69%	3.49%	42.68%
	Cr	90.50%	4.53%	2.01%	1.08%	0.74%	0.51%	0.37%	0.27%	61.99%	0.70%	1.13%	3.87%	32.30%
LP	Cb	89.19%	4.85%	2.31%	1.30%	0.84%	0.63%	0.46%	0.41%	50.32%	1.44%	1.24%	3.12%	43.88%
	Cr	90.11%	4.62%	2.01%	1.18%	0.72%	0.56%	0.42%	0.38%	60.39%	1.13%	1.39%	3.88%	33.22%
RA	Cb	77.65%	5.59%	3.43%	2.60%	2.12%	2.20%	2.63%	3.79%	58.38%	2.65%	1.36%	2.96%	34.64%
	Cr	78.78%	5.86%	3.12%	2.33%	2.02%	1.94%	2.39%	3.57%	54.18%	2.05%	1.75%	3.33%	38.70%

ratio of transform tools for the residual signal of intra-/inter-predicted CUs of the chroma components, respectively.

In the case of intra-predicted CU, both the luma and chroma components show similar ratios on the AI and RA configurations and on the LB and LP configurations, respectively. For the luma component, DST-VII is selected as the horizontal kernel in all configurations. For the vertical kernels, DCT-VIII is the most selected on the AI and RA configurations, and DST-VII is the most selected on LB and LP configurations, respectively. Moreover, determining the transform kernel by explicit index signaling is less selective than determining the transform kernel implicitly in all cases. In particular, DCT-VIII for the horizontal kernel on the AI and RA configurations is rarely selected, so it is expected that coding efficiency will not be affected, even if it is skipped from the encoding process. In the case of LFNST, which performs the secondary transform on the primary transform coefficients for the residual signal of intra-predicted CUs, it is applied only to the AI and RA configurations by JVET CTC and is applied to 18.31% and 15.79%, respectively.

In the chroma component, most regions are skipped on the LB and LP configurations, unlike the luma component. Furthermore, LFNST is applied to 20.55% and 2.42% on the AI and RA configurations, respectively. Considering the selection ratio of intra and inter prediction in Table 6, LFNST is analyzed as a technique that applies to approximately 2% of the luma component in the RA environments but shows a coding efficiency of approximately 0.7% [34].

In contrast to the intra-predicted CU, DST-VII and DCT-VIII kernels can be applied only when SBT is used for the inter-predicted CU, and the vertical and horizontal transform kernels of the non-SBT TU only apply DCT-II according to JVET CTC. As shown in Tables 20 and 21, the transform is skipped for most regions for both color components, and more than 90% is skipped, especially for the chroma components. Therefore, the encoding complexity will be improved if a strong early termination condition is added to the chroma components for the H.266/VVC encoder. In addition, SBT is applied only in a small region with 3.22%, 2.99%, and 1.73% on the LB, LP, and RA configurations, respectively. More specifically, when only 1/2 or 1/4 region of the transform is performed through the horizontal or vertical division of the TU, only the left subblock or the above subblock is transformed was the most often selected.

4.5 In-loop Filtering Tools

Table 22 presents the selection ratio for SAO among in-loop filtering tools in the luma and chroma components. SAO is applied to almost all blocks in the luma component on the AI configuration but not for at least 65% of the regions in the luma component on the LB, LP, and RA configurations. The chroma components also show a similar tendency, and it appears that the proportion not applied is larger than that of the luma component. In particular, the selection ratio for each category of SAO is applied the most to the horizontal edge, especially within the intra-predicted region.

Tables 23 and 24 show the performed proportion in the chroma and luma components for ALF and CC-ALF, which have achieved the highest coding efficiency among each newly adopted tool of H.266/VVC [34]. For the luma

Table 24. Ratios of ALF for luma components.

	ALF (Luma)		
	Off	Filter set index	
		Offline trained set (1~16)	Others (17~26)
AI	0.75%	99.25%	0.00%
LB	24.50%	38.75%	36.76%
LP	22.56%	39.29%	38.16%
RA	15.51%	53.10%	31.39%

Table 25. Experimental results of setting the CTU size to 64×64 (Anchor: 128×128).

	AI					RA				
	Y	Cb	Cr	EncT	DecT	Y	Cb	Cr	EncT	DecT
A1	0.40%	0.67%	1.39%	97%	102%	2.25%	2.29%	3.51%	84%	101%
A2	0.63%	0.94%	0.38%	96%	102%	3.50%	3.78%	4.78%	82%	100%
B	0.39%	0.14%	0.93%	97%	100%	1.78%	3.69%	4.61%	87%	102%
C	0.30%	0.34%	0.72%	97%	99%	0.32%	1.61%	2.18%	95%	101%
E	0.40%	-0.25%	2.30%	97%	100%	-	-	-	-	-
All	0.41%	0.34%	1.09%	97%	101%	1.83%	2.87%	3.77%	87%	101%
D	0.12%	-0.07%	0.46%	97%	100%	0.16%	0.61%	0.97%	99%	100%
F	-1.17%	-1.45%	-1.06%	85%	101%	0.46%	0.68%	1.48%	84%	105%

Table 26. Experimental results of setting the CTU size to 32×32 (Anchor: 128×128).

	AI					RA				
	Y	Cb	Cr	EncT	DecT	Y	Cb	Cr	EncT	DecT
A1	1.83%	6.39%	9.58%	86%	129%	14.00%	17.84%	23.05%	43%	129%
A2	2.43%	12.64%	5.94%	89%	126%	16.15%	21.90%	20.98%	44%	126%
B	1.59%	7.78%	11.71%	93%	121%	9.75%	18.58%	20.07%	48%	130%
C	0.85%	3.03%	3.95%	94%	111%	3.37%	6.87%	8.01%	59%	120%
E	2.33%	10.39%	14.32%	93%	125%	-	-	-	-	-
All	1.73%	7.73%	9.10%	91%	122%	10.18%	15.97%	17.63%	49%	126%
D	0.36%	1.16%	1.76%	93%	112%	2.02%	3.96%	4.99%	73%	116%
F	0.10%	2.34%	4.53%	79%	120%	6.56%	9.96%	10.97%	59%	146%

component, ALF is applied for almost all regions on the AI configuration, and the most selected filter set is the 16th index filter set among pre-trained filter sets, with 91.96%. In addition, the selection ratio of ALF for the chroma components is not large except for the AI configuration. On the other hand, CC-ALF is applied to more than half of the entire region, and the 4th index filter is the most selected. Therefore, the encoding complexity will be reduced without coding performance loss if the search order is changed, considering the selection ratio in the RDO process of the H.266/VVC encoder.

4.6 Block Structure and Partitioning Schemes

This subsection reports the coding efficiency according to the CTU size and the results of the tool-off experiments on the BT, TT, and DT, which are newly adopted partitioning structures compared to H.265/HEVC. Tables 25 and 26 show the performance when set to 64×64 and 32×32 compared to the maximum CTU size of H.266/VVC (128×128).

The coding loss is observed gradually as the CTU size is set smaller. This tendency is shown mainly in classes A1 and A2 with a UHD 4K resolution. H.266/VVC uses a CTU size four times larger than H.265/HEVC, which inevitably increases the hardware area compared to H.265/HEVC when designing a hardware decoder. Therefore, if the CTU size is limited, considering the resolution of the compressed video, it is expected to reduce the hardware area without reducing the coding efficiency significantly. On the other hand, limiting the CTU size to 32×32 will be difficult because of significant coding performance degradation, as shown in Table 26. Table 27 shows the performance for the maxTB is set to 32 compared to 64 according to JVET CTC. Because the intra prediction of H.266/VVC is performed per TU, setting maxTB to 32 reduces the maximum block size of intra prediction. Therefore, more block partitionings are performed in each intra-predicted CU. In addition, coding

Table 27. Experimental results of the setting maxTU to 32 (Anchor: 64).

	AI					RA				
	Y	Cb	Cr	EncT	DecT	Y	Cb	Cr	EncT	DecT
A1	0.56%	2.66%	2.60%	95%	107%	4.79%	7.54%	8.63%	93%	109%
A2	0.56%	2.71%	1.74%	96%	104%	1.22%	2.30%	1.71%	96%	103%
B	0.28%	2.96%	3.39%	97%	102%	0.92%	2.43%	3.05%	97%	104%
C	0.07%	0.46%	0.24%	98%	97%	0.27%	0.39%	0.43%	100%	101%
E	0.52%	3.42%	3.42%	97%	101%	-	-	-	-	-
All	0.37%	2.39%	2.29%	97%	102%	1.58%	2.88%	3.20%	97%	104%
D	0.05%	0.40%	0.19%	98%	97%	0.02%	0.52%	0.15%	102%	100%
F	0.16%	1.14%	1.14%	102%	101%	0.33%	0.86%	1.07%	100%	105%

Table 28. Experimental results of BT off (Anchor: VTM12.0).

	AI					RA				
	Y	Cb	Cr	EncT	DecT	Y	Cb	Cr	EncT	DecT
A1	3.97%	4.55%	5.27%	31%	97%	5.71%	6.07%	7.47%	36%	100%
A2	2.82%	7.99%	6.82%	29%	98%	5.76%	8.65%	7.77%	43%	104%
B	3.51%	7.65%	8.65%	28%	97%	5.93%	9.69%	10.62%	41%	104%
C	4.98%	6.98%	7.68%	23%	94%	6.49%	8.98%	10.33%	38%	101%
E	5.04%	7.32%	7.59%	29%	97%	-	-	-	-	-
All	4.06%	6.99%	7.39%	28%	97%	6.00%	8.57%	9.34%	40%	102%
D	4.86%	7.48%	7.71%	23%	94%	6.94%	10.99%	11.55%	41%	102%
F	7.12%	8.99%	9.47%	31%	97%	8.24%	9.75%	10.27%	44%	104%

Table 29. Experimental results of TT off (Anchor: VTM12.0).

	AI					RA				
	Y	Cb	Cr	EncT	DecT	Y	Cb	Cr	EncT	DecT
A1	0.95%	1.52%	1.60%	56%	97%	2.82%	3.68%	4.28%	48%	103%
A2	1.00%	1.66%	1.46%	51%	97%	2.80%	3.51%	3.25%	55%	101%
B	1.16%	1.59%	1.66%	51%	95%	2.70%	3.17%	3.52%	54%	100%
C	1.61%	2.28%	2.39%	47%	94%	3.37%	4.38%	4.48%	49%	99%
E	1.86%	1.37%	1.73%	52%	95%	-	-	-	-	-
All	1.32%	1.71%	1.79%	43%	96%	2.92%	3.66%	3.87%	51%	100%
D	1.49%	2.21%	2.65%	47%	95%	3.29%	4.82%	4.66%	54%	98%
F	2.38%	2.84%	2.92%	53%	96%	3.80%	4.26%	4.86%	57%	101%

loss is observed due to an increase in signaling information transmitted in each TU, such as cbfs. This tendency is shown mainly in classes A1 and A2, similar to the result of adjusting the CTU size, as shown in Tables 25 and 26. These results show that H.266/VVC obtained better coding efficiency by increasing the CTU size and maximum TU size in the block structure, in addition to newly adopting the coding techniques compared to H.265/HEVC.

Tables 28-31 show the results of the tool-off of BT, TT, and DT, respectively. Although BT consumes more complexity in the encoding process than TT, the coding efficiency is better. When BT and TT are turned off to use the same block partitioning structure as H.265/HEVC, about 8% coding loss is observed in each configuration, but the encoding complexity decreases by up to 10 times compared to the VTM12.0 encoder. These experimental results are obtained because coding techniques are searched recursively for different partitioning schemes in the video encoding process. Although the coding efficiency of BT, TT, and DT is not known precisely, H.266/VVC obtains a significant compression efficiency compared to H.265/HEVC because of the BT, TT, and DT, as shown in Tables 28-31.

4.7 Group of Pictures Structure

In addition to the block structure and partitioning scheme analysis in the previous subsection, the coding efficiency according to the number of reference pictures constituting the GOP in the RA configuration is analyzed comparatively.

H.265/HEVC and H.266/VVC use a hierarchical GOP

Table 30. Experimental results of both BT and TT off (Anchor: VTM12.0).

	AI					RA				
	Y	Cb	Cr	EncT	DecT	Y	Cb	Cr	EncT	DecT
A1	9.14%	9.72%	11.43%	12%	90%	15.33%	16.54%	19.04%	12%	93%
A2	5.15%	14.35%	12.02%	10%	93%	13.04%	19.18%	17.13%	18%	101%
B	6.38%	13.96%	15.62%	10%	91%	11.41%	18.92%	19.80%	18%	101%
C	9.39%	12.79%	13.96%	8%	85%	13.26%	18.10%	19.59%	17%	97%
E	10.35%	13.42%	13.30%	12%	91%	-	-	-	-	-
All	7.97%	12.97%	13.57%	10%	90%	13.01%	18.28%	19.06%	16%	99%
D	8.80%	12.94%	13.13%	9%	86%	12.60%	19.56%	20.23%	23%	97%
F	16.68%	19.68%	21.13%	11%	89%	20.75%	23.49%	24.88%	25%	102%

Table 31. Experimental results of DT off (Anchor: VTM12.0).

	AI					RA				
	Y	Cb	Cr	EncT	DecT	Y	Cb	Cr	EncT	DecT
A1	0.40%	0.67%	1.39%	97%	102%	0.02%	0.87%	0.16%	102%	102%
A2	0.63%	0.94%	0.38%	96%	102%	0.08%	3.40%	2.05%	101%	102%
B	0.39%	0.14%	0.93%	97%	100%	0.15%	5.04%	7.29%	102%	101%
C	0.30%	0.34%	0.72%	97%	99%	0.16%	2.20%	2.74%	101%	100%
E	0.40%	-0.25%	2.30%	97%	100%	-	-	-	-	-
All	0.41%	0.34%	1.09%	97%	101%	0.12%	3.12%	3.60%	102%	101%
D	0.12%	-0.07%	0.46%	97%	100%	0.16%	3.29%	3.85%	102%	100%
F	-1.11%	-1.37%	-0.98%	82%	101%	0.10%	2.20%	3.30%	105%	102%

Table 32. Experimental results of setting the GOP size to 16 on the RA configuration (Anchor: GOP 32).

	RA				
	Y	Cb	Cr	EncT	DecT
A1	2.00%	3.43%	4.46%	101%	102%
A2	3.29%	6.52%	5.81%	101%	100%
B	4.36%	7.79%	8.55%	102%	102%
C	3.61%	6.41%	6.72%	101%	100%
All	3.47%	6.30%	6.70%	101%	101%
D	3.51%	6.99%	7.06%	100%	100%
F	6.97%	8.02%	7.87%	105%	100%

structure when performing compression on the RA configuration according to the CTC, and the number of reference pictures that constitute the GOP (GOP size) is set to 16 and 32, respectively. Therefore, the coding efficiencies for the GOP size of 16 and 32 are compared, as shown in Table 32, to assess the coding efficiency of H.266/VVC and H.265/HEVC on the RA configuration fairly. As shown in Table 32, when the GOP size is set to 16, a coding loss of approximately 3.5% occurred in the luma component. This result is attributed to when the GOP is size set to 32, a reference picture with a higher temporal ID is inserted between each picture in the GOP of size 16, and the sliceQP of the inserted pictures is set to a high QP during encoding. As a result, the coding efficiency of H.266/VVC compared to H.265/HEVC on the RA configuration is obtained from both the newly adopted tools and in the GOP size.

5. Conclusion

H.266/VVC is the state-of-the-art video coding standard to facilitate efficient compression for a wide range of video content and services, such as UHD (8K or higher resolution), HDR/WCG video, screen content, and 360-degree videos, with approximately 40% better coding efficiency than H.265/HEVC. This paper addressed the modified or newly adopted techniques including in the coding tools for each module of H.266/VVC, in detail, by a comparison with H.265/HEVC. In addition, statistical analysis of the various coding techniques to be preceded for acceleration, optimization, and parallelization was conducted to assess the substantial computational complexity of the encoder. Bitstreams were generated using the VTM12.0 encoder for each configuration under the JVET CTC, and the statistical analyses of H.266/VVC coding tools and the coding performance of the block

Table 33. Notations used in Section 3.

Notation	Equation	Definition
P_{inter}	(3.3.1)	Inter-predicted signal using merge mode
P_{intra}		Intra-predicted signal using planar mode
W_{inter}		Weight for inter prediction
W_{intra}		Weight for intra prediction
P_{Lk}	(3.3.2)	Inter-predicted signal by the motion vector from list k ($k=0,1$)
w		Weight determined by signaling at the CU level
$mvdx_{Lk}, mvdy_{Lk}$	(3.3.3)	Motion vector difference for list k ($k=0,1$)
mv_{kx}, mv_{ky}	(3.3.4), (3.3.5)	Motion vectors for top-left ($k=0$), top-right ($k=1$), and bottom-left ($k=2$) corner control points
MV_k	(3.3.6)	Motion vector in the list k ($k=0,1$)
MV_{diff}		Motion vector offset
v_x, v_y	(3.3.7)-(3.3.9)	Remaining small displacement from I_c to I_0 (it is symmetrical to its motion from I_c to I_1)
$I_k(i,j)$	(3.3.9)	Discrete sample array at the spatial coordinate (i,j) from list k ($k=0,1,c$ (c denotes current picture))
$g_x(i,j), g_y(i,j)$	(3.3.12), (3.3.13)	Horizontal and vertical gradients of the subblock prediction at position (i,j) in the current block
$F(u)$	(3.4.1), (3.4.2)	Transformed coefficient
$p(x)$		Original signal
$v_{u,x}$		Basis element of $N\times 1$ basis vector ($u,x=0,1,\ldots,N-1$), N is the transform size
QP	(3.5.1)	Quantization parameter
C	(3.5.2)	Transform coefficient
Δ	(3.5.2), (3.5.3)	Quantization step size
$level$	(3.5.3)	Quantized coefficient level
$I(x,y)$	(3.6.1)	Reconstructed sample at position (x,y)
b_i	(3.6.2)	Clipping parameter determined by a clipping index d_i and a sample bit depth BD
c_i		Derived value from the Wiener-Hopf equation
$binCW[i]$	(3.6.3)	Number of mapped code values for each piece i
$OrgCW$		Value of uniformly sampled with 16 pieces depending on input codeword range
$deltaCRS$	(3.6.4)	Chroma scaling offset value
$r_{i,j}$	(3.7.1), (3.7.2)	Intra-predicted residual signal of a block at a position (i,j)
$Q(\cdot)$		Quantization operation

partitioning structure and GOP structure were performed. H.266/VVC provided significantly improved coding efficiency, but the increased computational complexity remains a problem to be solved in the future. The statistical analysis provided in this paper will promote more research for the real-time implementation of H.266/VVC and its successful commercialization in the future.

Acknowledgment

The present research has been partly conducted by the Research Grant of Kwangwoon University in 2022.

References

[1] V. Cisco, "Cisco visual networking index: Forecast and trends, 2017-2022," White Paper, 2019. Article (CrossRef Link)

[2] U. Cisco, "Cisco annual internet report (2018-2023)," White Paper, 2020. Article (CrossRef Link)

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 12, pp. 1649-1668, Dec. 2012. Article (CrossRef Link)

[4] Versatile Video Coding, Recommendation ITU-T H.266 and ISO/IEC 23090-3 (VVC), ITU-T and ISO/IEC JTC 1, Jul. 2020. Article (CrossRef Link)

[5] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, 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, Jul. 2003. Article (CrossRef Link)

[6] F. Bossen, X. Li, K. Suehring, K. Sharman, V.Seregin, and A. Tourapis, "AHG report: Test model software development (AHG3)," *JVET document V0003*, Apr. 2021. Article (CrossRef Link)

[7] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, "Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC)," *Proceedings of the IEEE*, Vol. 109, No. 9, pp. 1463-1493, Sep. 2021. Article (CrossRef Link)

[8] W. Hamidouche, T. Biatek, M. Abdoli, E. Francois, F. Pescador, M. Radosavljevic, D. Menard, and M. Raulet "Versatile video coding standard: A review from coding tools to consumers deployment," *IEEE Consumer Electronic Magazine*, Vol. 11, No. 5, pp. 10-24, Sep. 2022. Article (CrossRef Link)

[9] 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*, Vol. 31, No. 10, pp. 3736-3764, Oct. 2021. Article (CrossRef Link)

[10] VTM12.0, Article (CrossRef Link)

[11] ITU-T H.261, "Video codec for audiovisual services at p x 64 kbits/s," Nov. 1988. Article (CrossRef Link)

[12] ITU-T H.262, "Information technology - generic coding of moving pictures and associated audio information - Part 2: Video," July 1995. Article (CrossRef Link)

[13] ITU-T H.263, "Video coding for low bit rate communication," July 1996. Article (CrossRef Link)

[14] I.-K. Kim, J. Min, T. Lee, W.-J. Han, and J. Park, "Block partitioning structure in the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 12, pp. 1697-1706, Dec. 2012. Article (CrossRef Link)

[15] M. Budagavi, A. Fuldseth, G. Bjøntegaard, V. Sze and M. Sadafale, "Core transform design in the high efficiency video coding (HEVC) standard," *in IEEE Journal of Selected Topics in Signal Processing*, Vol. 7, No. 6, pp. 1029-1041, Dec. 2013. Article (CrossRef Link)

[16] J. Stankowski, C. Korzeniewski, M. Domanski, and T. Grajek, ''Ratedistortion optimized quantization in HEVC: Performance limitations,'' *In Proceedings of the IEEE Picture Coding Symposium (PCS)*, pp. 85-89, May 2015. Article (CrossRef Link)

[17] L. Wang, S. Hong, and K. Panusopone, "Gradual decoding refresh with virtual boundary," *In Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp.2129-2133, Sep. 2021. Article (CrossRef Link)

[18] J. Lee, J. Park, H. Choi, J. Byeon, and D. Sim, "Overview of VVC," *Broadcasting and Media Magazine*, Vol.24, No.4, pp. 10-25, Oct. 2019. Article (CrossRef Link)

[19] Y.-K. Wang, R. Skupin, M. Hannuksela, S. Deshpande, Hendary, V. Drugeon, R. Sjöberg, B. Choi, V. Seregin, Y. Sanchez, J. M. Boyce, W. Wan, and G. J. Sullivan, "The high-level syntax of the versatile video coding (VVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 10, pp. 3779-3800, Oct. 2021. Article (CrossRef Link)

[20] Y.-W. Huang, J. An, H. Haung, X. Li, S.-T 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*, Vol. 31, No. 10, pp. 3818-3833, Oct. 2021. Article (CrossRef Link)

[21] J. Pfaff, A. Filippov, S. Liu, X. Zhao, J. Chen, S. D.-L. Hernández, T. Wiegand, V. Rufitskiy, A. K. Ramasubramonian, and G. V. Auwera, "Intra prediction and mode coding in VVC," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 10, pp. 3834-3847, Oct. 2021. Article (CrossRef Link)

[22] H. Yang, H. Chen, J. Chen, S. Esenlik, S. Sethuraman, X. Xiu, E. Alshina, and J. Luo, "Subblock-based motion derivation and inter prediction refinement in the versatile video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 10, pp. 3862-3877, Oct. 2021. Article (CrossRef Link)

[23] W.-J. Chien, L. Zhang, M. Winken, X. Li, R.-L. Liao, H. Gao, C.-W. Hsu, H. Liu, and C.-C. Chen, "Motion vector coding and block merging in the versatile video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 10, pp. 3848-3861, Oct. 2021. Article (CrossRef Link)

[24] 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*, Vol. 31, No. 10, pp. 3878-3890, Oct. 2021. Article (CrossRef Link)

[25] H. Schwarz, M. Coban, M. Karczewicz, T.-D. 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*, Vol. 31, No. 10, pp. 3891-3906, Oct. 2021. Article (CrossRef Link)

[26] M. Karczewicz, N. Hu, J. Taquet, C.-Y. Chen, K. Misra, K. Andersson, P. Yin, T. Lu, E. François, and J. Chen, "VVC in-loop filters," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 10, pp. 3907-3925, Oct. 2021. Article (CrossRef Link)

[27] K. Ugur, A. Alshin, E. Alshina, F. Bossen, W.-J. Han, J.-H. Park, and J. Lainema, "Interpolation filter design in HEVC and its coding efficiency-complexity analysis," *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.26-31, May. 2013. Article (CrossRef Link)

[28] A. K. Ramasubramonian *et al.*, "CE3-1.6: On 1xN and 2xN subblocks of ISP," *JVET document O0106*, Jul. 2019. Article (CrossRef Link)

[29] 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 J. ICT Discoveries*, Vol. 3, No. 1, May 2020. Article (CrossRef Link)

[30] J. Huo, Y. Ma, S. Wan, Y. Yu, M. Wang, K. Zhang, L. Zhang, H. Liu, J. Xu, Y. Wang, J. Li, S. Wang, and W. Gao, "CE3-1.5: CCLM derived from four neighbouring samples," *JVET document N0271*, Apr. 2019. Article (CrossRef Link)

[31] J.-L. Lin, Y.-W. Chen, Y.-W. Huang, and S.-M. Lei, "Motion vector coding in the HEVC standard," *IEEE Journal of selected topics in Signal Processing*, Vol. 7, No. 6, pp. 957-968, Dec. 2013. Article (CrossRef Link)

[32] H. J. Song, Y.-L. Lee, "Inverse transform using linearity for video coding," *Electronics*, Vol. 11, No. 760, pp. 1-14, Dec. 2022. Article (CrossRef Link)

[33] Haykin, Simon S. "Adaptive filter theory," *Pearson Education India*, 2008. Article (CrossRef Link)

[34] W.-J. Chien, J. Boyce, Y.-W. Chen, R. Chernyak, K. Choi, R. Hashimoto, Y.-W. Huang, H. Jang, R.-L. Liao, and S. Liu, "JVET AHG report: tool reporting procedure and testing (AHG13)," *JVET document T0013*, Dec. 2020. Article (CrossRef Link)

[35] M. Karczewicz, N. Shlyakhov, N. Hu, V. Seregin, and W.-J. Chien, "CE2.4.1.4: reduced filter shape size for ALF," *JVET document K0371*, Jul. 2018. Article (CrossRef Link)

[36] C.-Y. Tsai, C.-Y. Chen, T. Yamakage, I. S. Chong, Y.-W. Huang, C.-M. Fu, T. Itoh, T. Watanebe, T. Chujoh, M. Karczewicz, and S.-M. Lei, "Adaptive loop filtering for video coding," *IEEE Journal of Selected Topics in Signal Processing*, Vol. 7, No. 6, pp. 934-945, Dec. 2013. Article (CrossRef Link)

[37] J. Taquet, P. Onno, C. Gisquet, and G. Laroche, "CE5: Results of Tests CE5-3.1, CE5-3.2, CE5-3.3 and CE5-3.4 on non-linear adaptive loop filter," *JVET document N0242*, Apr. 2019. Article (CrossRef Link)

[38] M. Karczewicz, L. Zhang, W.-J. Chien, and X. Li, "EE2.5: improvements on adaptive loop filter," *JVET document C0038*, Jun. 2016. Article (CrossRef Link)

[39] J. Taquet, P. Onno, C. Giquet, and G. Laroche, "CE5-4: alternative luma filter sets and alternative chroma filters for ALF," *JVET document O0090*, Jul. 2019. Article (CrossRef Link)

[40] A. M. Kotra, S. Esenlik, B. Wang, H. Gao, and J. Chen, "non-CE: loop filter line buffer reduction," *JVET document M0301*, Jan. 2019. Article (CrossRef Link)

[41] S.-C. Lim, J. Kang, H. Lee, J. Lee, and H. Y. Kim, "CE2: subsampled Laplacian calculation (Test 6.1, 6.2, 6.3, and 6.4)," *JVET document L0147*, Oct. 2018. Article (CrossRef Link)

[42] N. Hu, V. Seregin, H. E. Egilmez, and M. Karczewicz, "CE5: coding tree block based adaptive loop filter (CE5-4)," *JVET document N0415*, Mar. 2019. Article (CrossRef Link)

[43] K. Misra, F. Bossen, and A. Segall, "cross-component adaptive loop filter for chroma," *JVET document O0636*, Jul. 2019. Article (CrossRef Link)

[44] T. Lu, F. Pu, P. Yin, S. McCarty, W. Husak, T. Chen, E. Francois, C. Chevance, F. Hiron, J. Chen, R.-L. Liao, Y. Ye, and J. Luo, "luma mapping with chroma scaling in versatile video coding," *in Proc. IEEE Data Compression Conference (DCC)*, pp. 193-202, Mar. 2020. Article (CrossRef Link)

[45] E. Francois, C. A. Segall, A. M. Tourapis, P. Yin, and D. Rusanovskyy, "High dynamic range video coding technology in responses to the joint call for proposals on video compression with capability beyond HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 30, No. 5, pp. 1253-1266, May 2020. Article (CrossRef Link)

[46] T. Lu, F. Pu, P. Yin, W. Husak, S. McCarthy, and T. Chen, "CE12: mapping functions (test CE12-1 and CE12-2)," *JVET document M0427*, Jan. 2019. Article (CrossRef Link)

[47] E. François, F. Galpin, K. Naser, and P. D. Lagrange, "AHG7/AHG15: signalling of corrective values for chroma residual scaling," *JVET document P0371*, Oct. 2019. Article (CrossRef Link)

[48] S. Liu, L. Wang, P. Wu, and H. Yang, "JVET AHG report: neural networks in video coding (AHG9)," *JVET document J0009*, Apr. 2018. Article (CrossRef Link)

[49] S. Liu, Y. M. Li, B. Choi, K. Kawamura, Y. Li, L. Wang, P. Wu, and H. Yang, "JVET AHG report: neural networks in video coding (AHG9)," *JVET document P0009*, Oct. 2019. Article (CrossRef Link)

[50] Y. Li, S. Liu, and K. Kawamura, "CE10: summary

report on neural network based filter for video coding," *JVET document O0030*, Jul. 2019. Article (CrossRef Link)

[51] E. Alshina, S. Liu, A. Segall, J. Chen, F. Galpin, J. Pfaff, S. S. Wang, Z. Wang, M. Wien, P. Wu, and J. Xu, "JVET AHG report: neural network-based video coding (AHG11)," *JVET document Y0011*, Jan. 2022. Article (CrossRef Link)

[52] J. Xu, R. Joshi, and R. A. Cohen, "Overview of the emerging HEVC screen content coding extension." *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 26, No. 1, pp. 50-62, Jan. 2016. Article (CrossRef Link)

[53] T. Nguyen, X. Xu, F. Henry, R.-L. Liao, M. G. Sarwer, M. Karczewicz, Y.-H. Chao, J. Xu, S. Liu, D. Marpe, and G. J. Sullivan, "Overview of the screen content support in VVC: applications, coding tools, and performance," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 31, No. 10, pp. 3801-3817, Oct. 2021. Article (CrossRef Link)

[54] Y. Lee, B. Kim, and B. Jeon, "Study of sub-pel block vector for Intra Block Copy," *In International Workshop on Advanced Imaging Technology (IWAIT) 2022*, Vol. 12177, pp. 253-258, Jan. 2022. Article (CrossRef Link)

[55] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, "VTM common test conditions and software reference configurations for SDR video," *JVET document T2010*, Oct. 2020. Article (CrossRef Link)

Minhun Lee received his B.S. degree in Mathematics and Electronic Engineering (double major) and M.S. degree in Electronic Engineering from Kwangwoon University, Seoul, South Korea, in 2019 and 2021, respectively, where he is currently pursuing a Ph.D. degree in Computer Engineering. His current research interests include video coding, screen content coding, 3D reconstruction, and computer vision.

HyeonJu Song received her B.S. degree in Computer Engineering from Sejong University, Seoul, South Korea, in 2021, where she is currently pursuing her M.S. degree. Her current research interests include image and video compression, image processing, deep learning, and future video coding technologies.

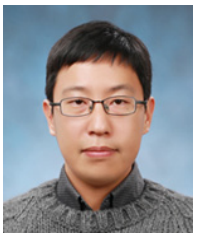
Jeeyoon Park received her B.S. degree in Computer Engineering from Kookmin University, Seoul, South Korea, in 2016. She is currently pursuing a Ph.D. degree in Electrical and Computer Engineering from Sungkyunkwan University, Suwon, South Korea. Her current research interests include image/video coding.

Byeungwoo Jeon received his B.S. degree (Magna Cum Laude) in 1985, M.S. degree in 1987 from the Department of Electronics Engineering, Seoul National University, Seoul, South Korea, and Ph.D. degree from the School of Electrical Engineering, Purdue University, West Lafayette, USA, in 1992. From 1993 to 1997, he was in the Signal Processing Laboratory, Samsung Electronics, Korea, where he worked on the research and development of video compression algorithms, the design of digital broadcasting satellite receivers, and other MPEG-related research for multimedia applications. Since September 1997, he has been at Sungkyunkwan University (SKKU), Korea, where he is currently a full professor. His research interests include multimedia signal processing, video compression, statistical pattern recognition, and remote sensing. He served as Project Manager of Digital TV and Broadcasting in the Korean Ministry of Information and Communications from 2004.3 to 2006.2, where he supervised all digital TV-related R&D in South Korea. From 2015.1 to 2016.12, he was the Dean of the College of Information and Communication Engineering, SKKU. In 2019, he was the President of the Korean Institute of Broadcast and Media Engineers. Dr. Jeon is a senior member of IEEE, a member of SPIE, an associate editor of IEEE Trans. on Broadcasting, and has been an associate editor of IEEE Trans. on Circuits and Systems for Video Technology. He was a recipient of the 2005 IEEK Haedong Paper Award from the Signal Processing Society in Korea, and the 2012 Special Service Award and 2019 Volunteer Award, both from the IEEE Broadcast Technology Society. In 2016, he was conferred Korean President's Commendation for his key role in promoting international standardization for video coding technology in South Korea.

Jungwon Kang received her B.S. and M.S. degrees in Electrical Engineering in 1993 and 1995, respectively, from Korea Aerospace University, Seoul, South Korea. She received her Ph.D. degree in Electrical and Computer Engineering in 2003 from the Georgia Institute of Technology, Atlanta, GA, US. Since 2003, she has been a principal member of the research staff in the Communication and Media Research Laboratory, ETRI, South Korea. She had made contributions to the HEVC and VVC standards. Her current research interests include video/image compression, immersive video processing, and multimedia applications.

Jae-Gon Kim received his B.S. degree in Electronics Engineering from Kyungpook National University, Daegu, South Korea, in 1990, the M.S. and Ph.D. degrees in Electronical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1992 and 2005, respectively. From 1992 to 2007, he was with the Electronics and Telecommunications Research Institute (ETRI), where he was involved in the development of digital broadcasting media services, MPEG-2/4/7/21 standards and related applications, and convergence media technologies. From 2001 to 2002, he was a Staff Associate at the Department of Electrical Engineering, Columbia University, New York, USA. Since 2007, he has been with the Korea Aerospace University, Goyang-si, Gyeonggi-do, Korea, where he is currently a professor in the School of Electronics and Information Engineering. From 2014 to 2015, he was a visiting scholar with the Video Signal Processing Lab. at the University of California, San Diego. He has been involved in video coding standards in JCT-VC and JVET activities of ITU-T VCEG and ISO/IEC MPEG. His research interests include digital video coding, video signaling processing, immersive video, digital broadcasting media, and multimedia applications.

Yung-Lyul Lee received his B.S. and M.S. degrees in Electronic Engineering from Sogang University, Seoul, South Korea, in 1985 and 1987, respectively, and his Ph.D. degree in Electrical and Electronic Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1999. He was a principal researcher at Samsung Electronics R&D center from 1987 to 2001. He has been a professor at the Department of Computer Engineering, Sejong University, Seoul, Korea, since 2001. He was a visiting scholar at the University of Texas at Arlington, USA, from September 2006 to August 2007. He had contribution documents to the AVC/H.264, High Efficiency Video Coding (HEVC/H.265), and Versatile Video Coding (VVC/H.266) standards and had patents in the AVC/H.264, HEVC/H.265, and VVC/H.266 standards. Prof. Lee received a Minister prize from the Ministry of Commerce, Industry and Energy in Korea in November 2006, and a Korea Science Technology Superiority paper prize in October 2006. He was the president of the Korea Institute of Broadcasting and Media Engineering. His current research interests include video compression, digital signal processing, image processing, 3D video coding, deep learning, and CNN-based video coding, object detection-based video coding.



Je-Won Kang received his B.S. and M.S. degrees in Electrical Engineering and Computer Science from Seoul National University, Seoul, South Korea, in 2006 and 2008, respectively, and his Ph.D. degree in Electrical Engineering from the University of Southern California, Los Angeles, CA, USA, 2012. He was a senior engineer with the Multimedia R&D and Standard team in Qualcomm Technologies, Inc., San Diego, CA, USA, from 2012 to 2014. He was a visiting researcher at the Tampere University and Nokia research center in Tampere, Finland, in 2011 and Mitsubishi Electric Research Lab. in Boston, USA, in 2010. He has been an active contributor to the recent international video coding standards in JCT-VC, including High Efficiency Video Coding (HEVC) standard and the extensions to multi-view videos, 3D videos, and screen content videos and published more than 100 technical papers in international journals and conferences. He was an APSIPA Distinguished Lecturer from 2021 to 2022. He served as an Associate Editor of APSIPA Transactions on Signal and Information Processing. He is an Associate Professor at Ewha Womans University, Seoul, South Korea, and the head of the Information Coding and Processing Lab at the Department of Electronics and Electrical Engineering. His current research interests include image and video processing and compression, computer vision, and machine learning.

Donggyu Sim received his B.S. and M.S. degrees in Electronic Engineering and his Ph.D. degree from Sogang University, South Korea, in 1993, 1995, and 1999, respectively. He was with the Hyundai Electronics Company Ltd. from 1999 to 2000, being involved in MPEG-7 standardization. He was a Senior Research Engineer with Varo Vision Company Ltd., working on MPEG-4 wireless applications from 2000 to 2002. He worked at the Image Computing Systems Laboratory. (ICSL), University of Washington, as a Senior Research Engineer from 2002 to 2005. He researched ultrasound image analysis and parametric video coding. Since 2005, he has been with the Department of Computer Engineering, Kwangwoon University, Seoul, South Korea. In 2011, he joined Simon Frasier University as a Visiting Scholar. He is one of the leading inventors in many essential patents licensed to MPEG-LA for HEVC standard. His current research interests include video coding, video processing, computer vision, and video communication.