# Overview and Extensions of the High Efficiency Video Coding (HEVC) and Beyond (Versatile Video Coding)

**Research** · December 2019

DOI: 10.13140/RG.2.2.11353.26726



**2 authors:**

Shreyanka Subbarayappa
M. S. Ramaiah University of Applied Sciences
**8** PUBLICATIONS **12** CITATIONS

SEE PROFILE

Kamisetty Rao
University of Texas at Arlington
**186** PUBLICATIONS **10,235** CITATIONS

SEE PROFILE

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

# Overview and Extensions of the High Efficiency Video Coding (HEVC) and Beyond (Versatile Video Coding)

Shreyanka Subbarayappa[1], K.R.Rao[2]

[1]*Assistant Professor, E.C.E. Dept., Ramaiah University of Applied Sciences, Bangalore, KA 560054, India*
[2]*Professor, E.E. Dept., University of Texas at Arlington, Arlington, TX 76019, U.S.A.*

***Abstract*— High Efficiency Video Coding (HEVC) is the latest Video Coding standard by ITU-T and ISO organizations [1]. It challenges the state-of-the-art H.264/AVC Video Coding standard which is in current use in the industry and conquering 82% of market by 2018, by being able to reduce the bit rate by 50%, retaining the same video quality as of H.264 [26]. This paper provides overview of technical features and describes extensions of the HEVC standard along with the results obtained by JM and HM software's. It also touches the next video codec being introduced by ITU-T and ISO standard known as H.266/VVC or Versatile Video Codec [73] by June 2020.**

***Keywords*—coding tree unit, HEVC, HM software, SAO, VVC.**

## I. INTRODUCTION

High Efficiency Video Coding (HEVC) is the latest Video Coding format [1]. It challenges the state-of-the-art H.264/AVC Video Coding standard which is in current use in the industry by being able to reduce the bit rate by 50%, retaining the same video quality. It came into existence in the early 2012 although Joint Collaborative Team on Video Coding (JCT-VC) was formed in January 2001 to carry out developments on HEVC, and ever since then a huge range of development has been going on. On 13 April 2013, HEVC standard also called H.265 was approved by ITU-T. Joint Collaborative Team on Video Coding (JCT-VC), is a group of video coding experts from ITU-T Study Group (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG).

HEVC is designed to address existing applications of H.264/MPEG-4 AVC and to focus on two key issues: increased video resolution and increased use of parallel processing architectures [1]. It primarily targets consumer applications as pixel formats are limited to 4:2:0 8-bit and 4:2:0 10-bit. Main, Main 10 and Main intra profile (Main Still Picture profile) were finalized in 2013 [1].

The next revision of the standard [72], in July 2014, had enabled new use-cases with the support of additional pixel formats such as 4:2:2 and 4:4:4 and bit depth higher than 10-bit [12], embedded bit-stream scalability and 3D video [2].

Multimedia consumer applications have a very large market [1,2]. The revenues involved in digital TV broadcasting and DVD, Blu-ray distributions are substantial. Thus, standardization of video coding is essential. Standards simplify inter-operability between encoders and decoders from different manufacturers, they make it possible for different vendors to build platforms that incorporate video codecs, audio codecs, security and rights management and they all interact in well-defined and consistent ways. There are numerous video compression standards, both open source and proprietary, depending on the applications and end-usage. Figure I shows the evolution of the video codec standards from the 90's till today.

An increasing diversity of services, the growing popularity of HD video, and the emergence of beyond-HD formats (e.g., 4k×2k or 8k×4k resolution) (Figure II) [25] are creating even stronger needs for coding efficiency superior to H.264/MPEG-4 AVC's capabilities. The need is even stronger when higher resolution is accompanied by stereo or multiview capture and display. Moreover, the traffic caused by video applications targeting mobile devices and tablet PCs, as well as the transmission needs for video-on-demand services, are imposing severe challenges on today's networks. An increased desire for higher quality and resolutions is also arising in mobile applications [1].

Storage space and bandwidth are limited. Even if high bandwidth technology (e.g. fiber-optic cable) was in place, the per-byte-cost of transmission would have to be very low before it would be feasible to use it for the staggering amounts of data required by HDTV and ultra HDTV.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

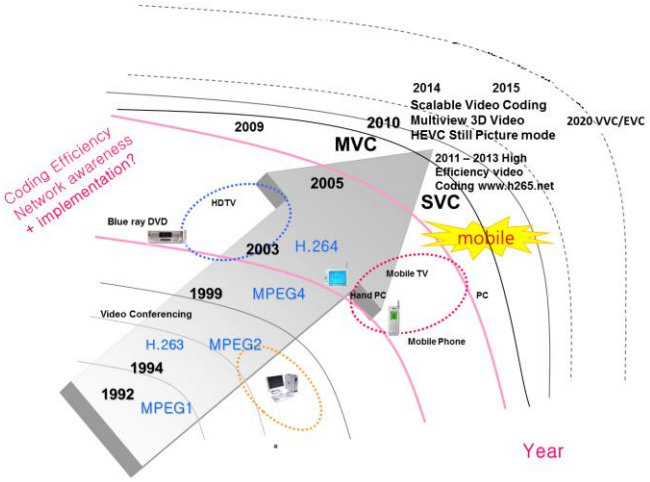



**Figure I Evolution Of Video Coding Standards [20]**


**FIGURE II The Spatial Resolutions Ranging From SD To Ultra HD Video**

## II. HEVC Coding Design

### A. Encoder and Decoder in HEVC

Since H.261, all the video compression standards have been employing hybrid approach for the video coding layer [34]. Similarly, HEVC does the same. Figure III shows the block diagram encoder to create a compressed video bit stream. The compressed bit stream is stored or transmitted. A video decoder decompresses the bit stream to create a sequence of decoded frames [13]. The video encoder performs the following process to generate the bitstream yielding to HEVC standard. Figures IV and V depict the block diagram of a hybrid video encoder and decoder, respectively. Each picture is partitioned into block shaped multiple units; the same exact partitioning is conveyed to the decoder [1]. After partitioning each picture is intra or inter Predicted.

The first picture of a video sequence and the first picture at each clean random-access point is coded using Intra prediction only. The rest of the pictures in a video sequence is coded using inter prediction. Linear spatial transformation is performed on the residual signal (the difference between the original picture unit and the prediction). The transform coefficients along with the prediction information are quantized and entropy encoded. In order to generate identical predictions for successive data the encoder clones the decoder processing loop [1]. The loop performs inverse scaling, inverse transform to construct residual signal which is then added to the prediction. It is further processed to remove the artifacts induced by block-wide processing and quantization. The decoded picture buffer stores the final picture representation for prediction of successive data. The video decoder decodes the encoded bitstream by performing the following steps: Entropy decoding and extracting the elements of the coded sequence. Rescaling and inverting the transform stage. Predicting each unit and adding the prediction.

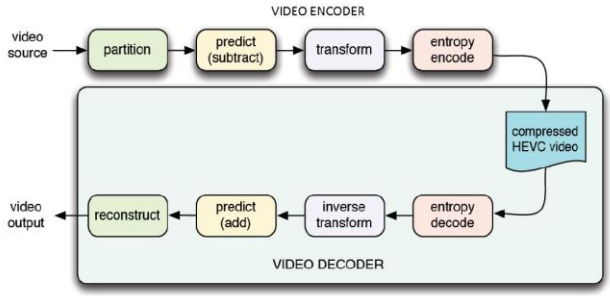



**FIGURE III HEVC Codec Block Diagram [13]**

### B. Features and coding techniques in HEVC

*1. Coding tree units and coding units*: HEVC supports highly flexible partitioning of a video sequence [1]. Each frame of the sequence is split into rectangular or square regions (units or blocks). The conventional macroblock is replaced by coding tree unit (CTU) in HEVC, which has a size selected by encoder and can be larger than a macroblock [1]. Each CTU consists of a luma CTB and the two chroma CTBs and syntax elements. In a CTU, a luma CTB has a picture area of $L \times L$ samples of the luma component and the corresponding chroma CTBs cover each $(L/2) \times (L/2)$ samples of each of the two chroma components. CTBs have always square shapes. The value of $L$ may be equal to 16, 32, or 64 as determined by an encoded syntax element specified in the sequence parameter set (SPS).

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

The value of *L* can be 16, 32 or 64. The larger CTBs are useful when encoding high-resolution video content and also enable better compression. See Refs. [35] and [36].

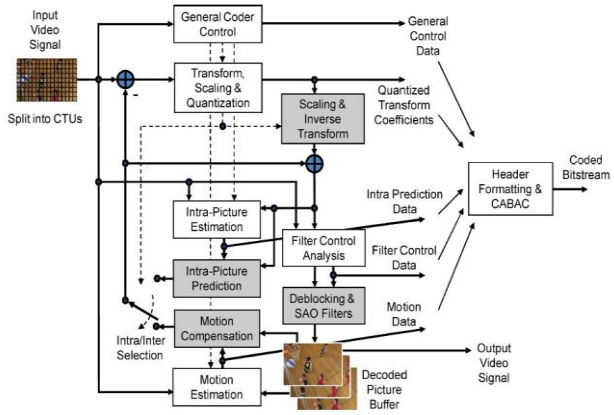



**FIGURE IV HEVC Video Encoder Block Diagram [1]**

Each CTB is split into one or multiple coding units (CUs). Each CU consists of one luma coding block (CB), two chroma CBs and associated syntax. Figure VI shows the structure of CTU and CU in a video frame. The minimum luma CB size is computed from *L* and the maximum depth of a quad-tree, and is always 8×8 or larger (in units of luma samples) [1]. Figure VII shows the quad-tree structure. The coding mode, intra or inter prediction, is selected at the CU level [1]. CBs have always square shapes. CU has an associated partitioning into prediction units (PUs) and transform units (TUs). A CU is the root for both prediction unit (PU) and transform unit (TU) as shown in Figure VIII. Figure IX shows examples of various CTU sizes and CU sizes suitable for different resolutions and types of content. For example, for an application using 1080p content that is known to include only simple global motion activities, a CTU size of 64 (*L*=64) and maximum depth of 2 may be appropriate choice. For more general 1080p content, which may also include complex motion activities of small regions, a CTU size of 64 and maximum depth of 4 would be preferable [21]. Pictorial representations of various block divisions for HEVC in a Frame is shown in Fig. X.

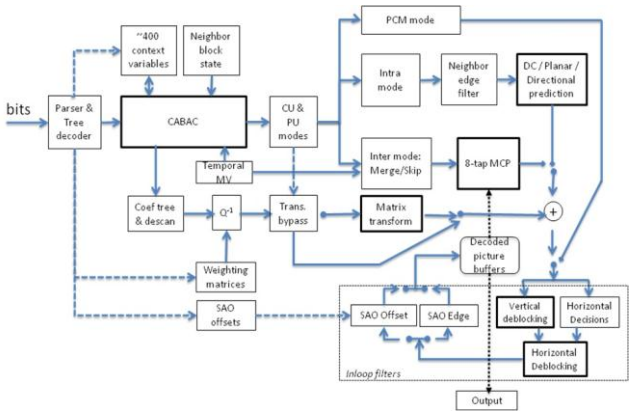



**FIGURE V HEVC Video Decoder Block Diagram [7]**

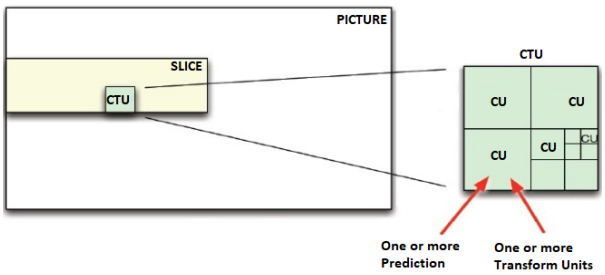



**FIGURE VI Coding Tree Unit (CTU), Coding Unit (CU) [13]**

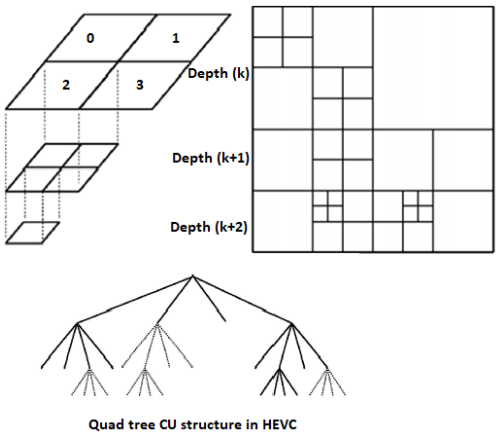



**FIGURE VII (a) QUAD TREE CU STRUCTURE IN HEVC [22]**

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

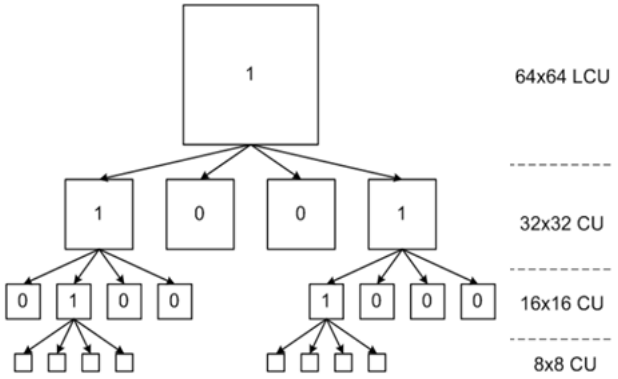



FIGURE VII (b) QUAD TREE SPLITTING FLAGS ARE 1'S AND 0'S [46]

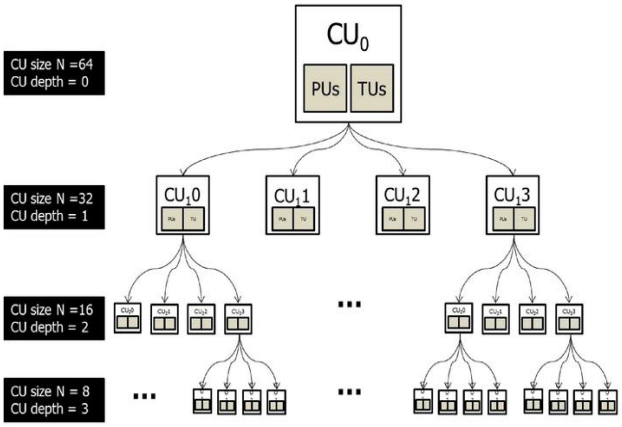



FIGURE VIII CODING STRUCTURE IN HEVC [44]

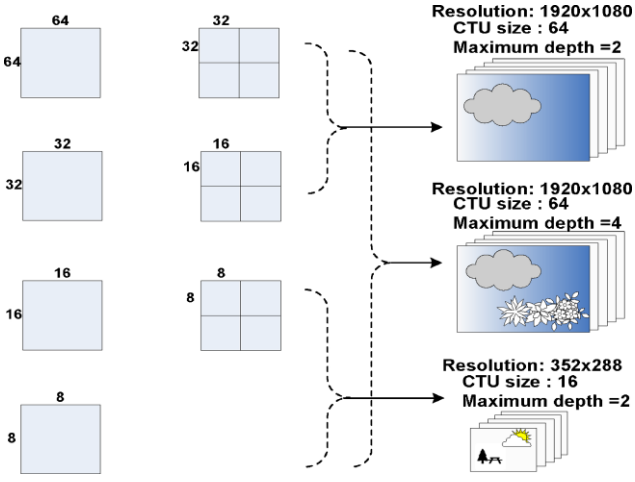



FIGURE IX FLEXIBLE CU PARTITIONING [21]

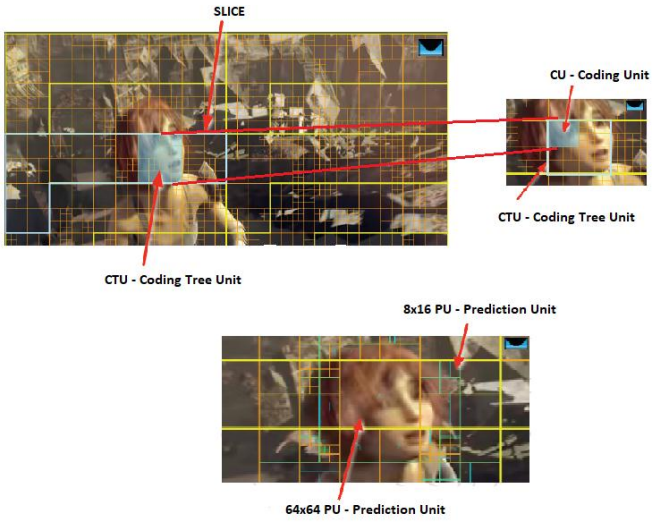



FIGURE X PICTORIAL REPRESENTATIONS OF VARIOUS BLOCK DIVISIONS FOR HEVC IN A FRAME [13]

*2. Prediction units:* The luma and chroma PBs, together with the associated prediction syntax, form the PU. The luma and chroma CBs are split into luma and chroma prediction blocks (PBs) based on prediction-type decision [1]. The prediction mode for the CU is signaled as being intra or inter, according to whether it uses intrapicture (spatial) prediction or interpicture (temporal) prediction. The size of PB can vary from 64×64 to 4×4 samples. For the prediction mode intra, except for the smallest CB size all PB size is same as the CB size that is allowed in the bitstream. Intra prediction can be performed on only square partitions. When CB has to be split into four PB which have their own intrapicture prediction mode, a flag is enabled. The reason for allowing this split is to enable distinct intrapicture prediction mode selections for blocks as small as 4×4 in size. When the luma intrapicture prediction operates with 4×4 blocks, the chroma intrapicture prediction also uses 4×4 blocks (each covering the same picture region as four 4×4 luma blocks). The actual region size at which the intrapicture prediction operates (which is distinct from the PB size, at which the intrapicture prediction mode is established) depends on the residual coding partitioning.

When the prediction mode is signaled as inter, it is specified whether the luma and chroma CBs are split into one, two, or four PBs. The splitting into four PBs is allowed only when the CB size is equal to the minimum allowed CB size, using an equivalent type of splitting as could otherwise be performed at the CB level of the design rather than at the PB level.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

When a CB is split into four PBs, each PB covers a quadrant of the CB. When a CB is split into two PBs, six types of this splitting are possible. The partitioning possibilities for interpicture-predicted CBs are depicted in Fig. XI. The upper partitions illustrate the cases of not splitting the CB of size M×M, of splitting the CB into two PBs of size M × (M/2) or (M/2) × M, or splitting it into four PBs of size (M/2) × (M/2). The lower four partition types in Fig. XI are referred to as asymmetric motion partitioning (AMP), and are only allowed when M is 16 or larger for luma. For intrapicture-predicted CBs, only M×M and (M/2) × (M/2) are supported.

To minimize worst-case memory bandwidth, PBs of luma size 4×4 are not allowed for interpicture prediction, and PBs of luma sizes 4×8 and 8×4 are restricted to unipredictive coding.

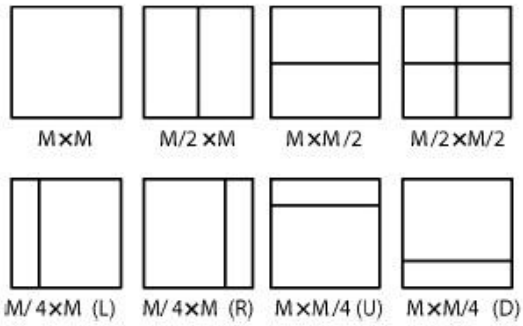



**FIGURE XI MODES FOR SPLITTING A CB INTO PBS [1]. L=LEFT, R=RIGHT, U=UP, D=DOWN**

*3. Transform units:* The prediction residual is coded using block transforms [1]. A TU tree structure has its root at the CU level. The luma CB residual may be identical to the luma transform block (TB) or may be further split into smaller luma TBs. The same applies to the chroma TBs. Integer basis functions similar to those of a discrete cosine transform (DCT) are defined for the square TB sizes 4×4, 8×8, 16×16, and 32×32. For the 4×4 transform of luma intrapicture prediction residuals, an integer transform derived from a form of discrete sine transform (DST) is alternatively specified.

Only square CB and TB partitioning is specified, where a block can be recursively split into quadrants, as illustrated in Fig. XII. For a given luma CB of size M×M, a flag signals whether it is split into four blocks of size M/2×M/2. If further splitting is possible, as signaled by a maximum depth of the residual quadtree indicated in the SPS, each quadrant is assigned a flag that indicates whether it is split into four quadrants.

The leaf node blocks resulting from the residual quadtree are the transform blocks that are further processed by transform coding. The encoder indicates the maximum and minimum luma TB sizes that it will use. Splitting is implicit when the CB size is larger than the maximum TB size. Not splitting is implicit when splitting would result in a luma TB size smaller than the indicated minimum.

The chroma TB size is half the luma TB size in each dimension, except when the luma TB size is 4×4, in which case a single 4×4 chroma TB is used for the region covered by four 4×4 luma TBs. In the case of intrapicture-predicted CUs, the decoded samples of the nearest-neighboring TBs (within or outside the CB) are used as reference data for intrapicture prediction. In contrast to previous standards, the HEVC design allows a TB to span across multiple PBs for interpicture-predicted CUs to maximize the potential coding efficiency benefits of the quadtree-structured TB partitioning.

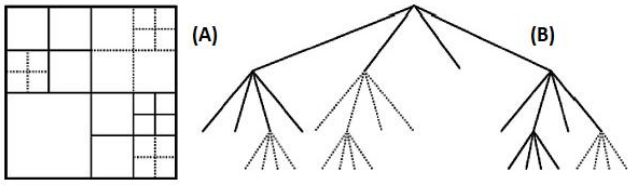



**FIGURE XII SUBDIVISION OF A CTB INTO CBS [AND TRANSFORM BLOCK (TBS)]. SOLID LINES INDICATE CB BOUNDARIES AND DOTTED LINES INDICATE TB BOUNDARIES. (A) CTB WITH ITS PARTITIONING. (B) CORRESPONDING QUADTREE [1].**

*4. Slices and Tiles:* Slices are processed in the order of a raster scan [1]. A picture may be split into one or several slices as shown in Fig. III (A) so that a picture is a collection of one or more slices. Slices are self-contained in the sense that, given the availability of the active sequence and picture parameter sets, their syntax elements can be parsed from the bitstream and the values of the samples in the area of the picture that the slice represents can be correctly decoded without the use of any data from other slices in the same picture. Tiles are self-contained and independently decodable rectangular regions of the picture. The main purpose of tiles is to enable the use of parallel processing architectures for encoding and decoding. Multiple tiles may share header information by being contained in the same slice. Alternatively, a single tile may contain multiple group of CTUs as shown in Fig. XIII(B).

*5. Intrapicture Prediction:* Intrapicture prediction operates according to the TB size, and previously decoded boundary samples from spatially neighboring TBs are used to form the prediction signal [1]. The possible prediction directions are shown in Fig. XIV.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

Directional prediction with 33 different directional orientations is defined for (square) TB sizes from 4×4 up to 32×32. Alternatively, planar prediction (assuming an amplitude surface with a horizontal and vertical slope derived from the boundaries) and DC prediction (a flat surface with a value matching the mean value of the boundary samples) can also be used. For chroma, the horizontal, vertical, planar, and DC prediction modes can be explicitly signaled, or the chroma prediction mode can be indicated to be the same as the luma prediction mode (and, as a special case to avoid redundant signaling, when one of the first four choices is indicated and is the same as the luma prediction mode, the Intra_Angular [34] mode is applied instead). The number of supported prediction modes varies based on the PU size (see Table I) [23]. From an encoding perspective, the increased number of prediction modes will require good mode selection heuristics to maintain a reasonable search complexity (see Table II) [24].

*6. Interpicture prediction:* Compared to intrapicture-predicted CBs, HEVC supports more PB partition shapes for interpicture-predicted CBs [1]. The partitioning modes of PART_2N×2N, PART_2N×N, and PART_N×2N indicate the cases when the CB is not split, split into two equal-size PBs horizontally, and split into two equal-size PBs vertically, respectively. Figure XV shows Intra and Inter frame prediction modes for HEVC.

*7. Motion vector signalling:* Advanced motion vector prediction (AMVP) is used, including derivation of several most probable candidates based on data from adjacent PBs and the reference picture [1]. A merge mode for MV coding can also be used, allowing the inheritance of MVs from temporally or spatially neighbouring PBs. Moreover, compared to H.264/MPEG-4 AVC, improved skipped and direct motion inferences are also specified.

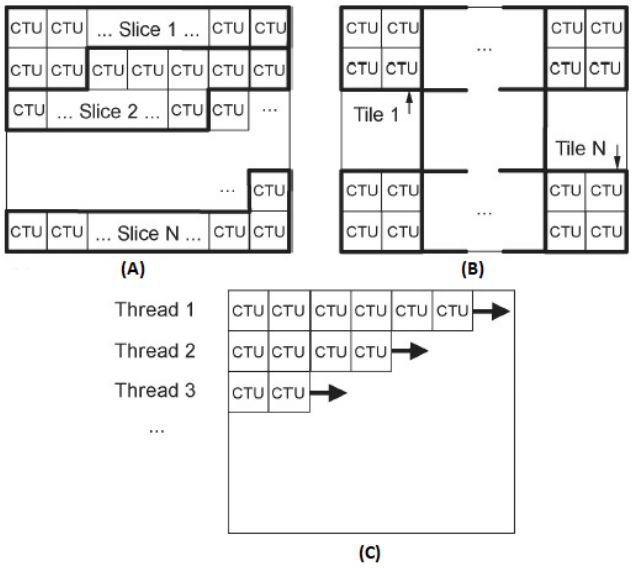



**FIGURE XIII SUBDIVISION OF A PICTURE INTO (A) SLICES AND (B) TILES. (C) ILLUSTRATION OF WAVE FRONT PARALLEL PROCESSING [1]**

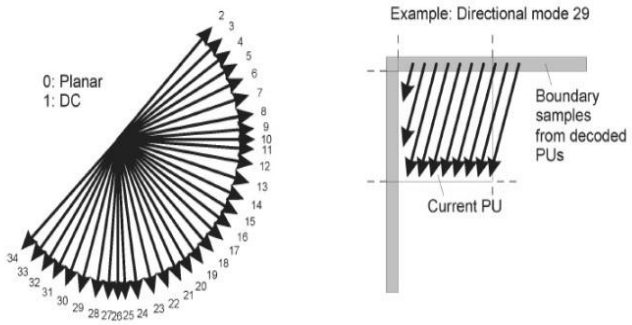



**FIGURE XIV MODES AND DIRECTIONAL ORIENTATIONS FOR INTRAPICTURE PREDICTION [1]**

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

TABLE I
LUMA INTRAPREDICTION MODES SUPPORTED BY DIFFERENT PU SIZES [23]

| PU Size | Intraprediction Modes | Number of Intra Prediction Modes |
|---|---|---|
| 64×64 | 0–2, 34 | 4 |
| 32×32 | 0–34 | 35 |
| 16×16 | 0–34 | 35 |
| 8×8 | 0–34 | 35 |
| 4×4 | 0–16, 34 | 18 |

TABLE II
TOTAL NUMBER OF MODES TO BE TESTED [47]

| Size of PB | Number of PBs in a 64×64 CU | Number of modes to be tested in each PB | Total number of modes to be tested at this level |
|---|---|---|---|
| 32×32 | 4 | 35 | 140 |
| 16×16 | 16 | 35 | 560 |
| 8×8 | 64 | 35 | 2240 |
| 4×4 | 256 | 18 | 4608 |
| | | Total | 7548 |

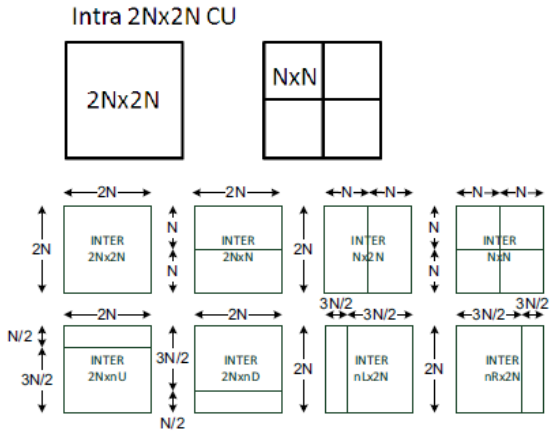



FIGURE XV INTRA AND INTER FRAME PREDICTION MODES FOR HEVC [5]

*8. Motion compensation:* Like MPEG-4/AVC [1], HEVC specifies motion vectors in 1/4-pel, but uses an 8-tap filter for luma (all positions), and a 4-tap 1/8-pel filter for chroma [1] as shown in Fig. XVI.

Because of the 8-tap filter, any given N×M sized block requires extra pixels on all sides (3 left/above, 4 right and below) to provide the filter with the data it needs. With small blocks like an 8×4, (8+7) × (4+7) = 15×11 pixels are needed. The HEVC standard limits the smallest block to be uni-directional and 4×4 is not supported since more small blocks require more memory read, thus increasing more memory access, more time and more power. The HEVC standard also supports weighted prediction for both uni- and bi-directional PUs. However, the weights are always explicitly transmitted in the slice header; there is no implicit weighted prediction like in MPEG-4/AVC [1]. Quarter-sample precision is used for the motion vectors. 7-tap (weights: -1, 4, -10, 58, 17, -5, 1) or 8-tap (weights: -1, 4, -11, 40, 40, -11, 4, 1) filters are used for interpolation of fractional-sample positions. The 8-tap filter is applied for half sample positions and the 7-tap filter is applied for quarter sample positions. Similar to H.264/MPEG-4 AVC [1], multiple reference pictures are used as shown in Fig. XVII. For each PB, either one or two motion vectors can be transmitted, resulting either in unipredictive or bipredictive coding, respectively. A scaling and offset operation can be applied to the prediction signal/signals in a manner known as weighted prediction.

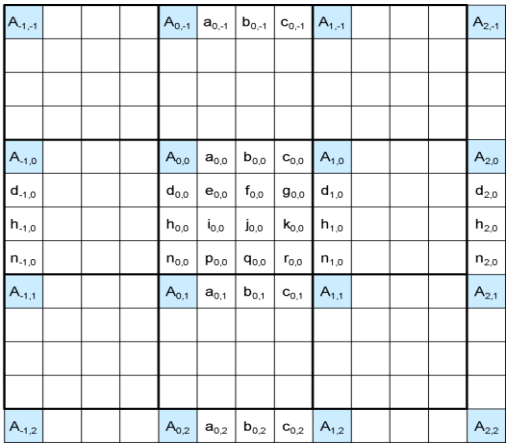



FIGURE XVI INTEGER AND FRACTIONAL SAMPLE POSITION FOR LUMA INTERPOLATION [1]

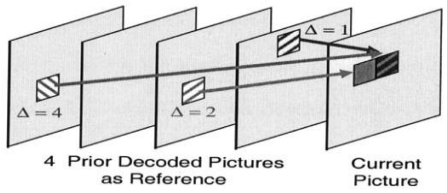



FIGURE XVII MULTIPLE PICTURES USED AS REFERENCE FOR THE CURRENT PICTURE FOR MOTION COMPENSATION [6]

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

*9. Transform, Scaling and Quantization:* HEVC uses transform coding of the prediction error residual in a similar manner as in prior standards [1]. The supported transform block sizes are 4×4, 8×8, 16×16, and 32×32.

Smaller size transform matrices are embedded in larger size transform matrices. This simplifies implementation, since a 32×32 matrix, can do 4×4, 8×8, 16×16, and 32×32 transform. Please see page 19 in [71]. Transform matrices 4×4 thru 32×32 INTDCTs (embedded) are shown. Transform matrices for each size transform are as follows.

**nS = 4**

{64, 64, 64, 64}
{83, 36,-36,-83}
{64,-64,-64, 64}
{36,-83, 83,-36}

**nS = 8**

{64, 64, 64, 64, 64, 64, 64, 64}
{89, 75, 50, 18,-18,-50,-75,-89}
{83, 36,-36,-83,-83,-36, 36, 83}
{75,-18,-89,-50, 50, 89, 18,-75}
{64,-64,-64, 64, 64,-64,-64, 64}
{50,-89, 18, 75,-75,-18, 89,-50}
{36,-83, 83,-36,-36, 83,-83, 36}
{18,-50, 75,-89, 89,-75, 50,-18}

**nS = 16**

{64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}
{90 87 80 70 57 43 25 9 -9-25-43-57-70-80-87-90}
{89 75 50 18-18-50-75-89-89-75-50-18 18 50 75 89}
{87 57 9-43-80-90-70-25 25 70 90 80 43 -9-57-87}
{83 36-36-83-83-36 36 83 83 36-36-83-83-36 36 83}
{80 9-70-87-25 57 90 43-43-90-57 25 87 70 -9-80}
{75-18-89-50 50 89 18-75-75 18 89 50-50-89-18 75}
{70-43-87 9 90 25-80-57 57 80-25-90 -9 87 43-70}
{64-64-64 64 64-64-64 64 64-64-64 64 64-64-64 64}
{57-80-25 90 -9-87 43 70-70-43 87 9-90 25 80-57}
{50-89 18 75-75-18 89-50-50 89-18-75 75 18-89 50}
{43-90 57 25-87 70 9-80 80 -9-70 87-25-57 90-43}
{36-83 83-36-36 83-83 36 36-83 83-36-36 83-83 36}
{25-70 90-80 43 9-57 87-87 57 -9-43 80-90 70-25}
{18-50 75-89 89-75 50-18-18 50-75 89-89 75-50 18}
{ 9-25 43-57 70-80 87-90 90-87 80-70 57-43 25 -9}

**nS = 32**

{64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64}
{90 90 88 85 82 78 73 67 61 54 46 38 31 22 13 4 -4-13-22-31-38-46-54-61-67-73-78-82-85-88-90-90}
{90 87 80 70 57 43 25 9 -9-25-43-57-70-80-87-90-90-87-80-70-57-43-25 -9 9 25 43 57 70 80 87 90}
{90 82 67 46 22 -4-31-54-73-85-90-88-78-61-38-13 13 38 61 78 88 90 85 73 54 31 4-22-46-67-82-90}
{89 75 50 18-18-50-75-89-89-75-50-18 18 50 75 89 89 75 50 18-18-50-75-89-89-75-50-18 18 50 75 89}
{88 67 31-13-54-82-90-78-46 -4 38 73 90 85 61 22-22-61-85-90-73-38 4 46 78 90 82 54 13-31-67-88}
{87 57 9-43-80-90-70-25 25 70 90 80 43 -9-57-87-87-57 -9 43 80 90 70 25-25-70-90-80-43 9 57 87}
{85 46-13-67-90-73-22 38 82 88 54 -4-61-90-78-31 31 78 90 61 4-54-88-82-38 22 73 90 67 13-46-85}
{83 36-36-83-83-36 36 83 83 36-36-83-83-36 36 83 83 36-36-83-83-36 36 83 83 36-36-83-83-36 36 83}
{82 22-54-90-61 13 78 85 31-46-90-67 4 73 88 38-38-88-73 -4 67 90 46-31-85-78-13 61 90 54-22-82}
{80 9-70-87-25 57 90 43-43-90-57 25 87 70 -9-80-80 -9 70 87 25-57-90-43 43 90 57-25-87-70 9 80}
{78 -4-82-73 13 85 67-22-88-61 31 90 54-38-90-46 46 90 38-54-90-31 61 88 22-67-85-13 73 82 4-78}
{75-18-89-50 50 89 18-75-75 18 89 50-50-89-18 75 75-18-89-50 50 89 18-75-75 18 89 50-50-89-18 75}
{73-31-90-22 78 67-38-90-13 82 61-46-88 -4 85 54-54-85 4 88 46-61-82 13 90 38-67-78 22 90 31-73}
{70-43-87 9 90 25-80-57 57 80-25-90 -9 87 43-70-70 43 87 -9-90-25 80 57-57-80 25 90 9-87-43 70}
{67-54-78 38 85-22-90 4 90 13-88-31 82 46-73-61 61 73-46-82 31 88-13-90 -4 90 22-85-38 78 54-67}
{64-64-64 64 64-64-64 64 64-64-64 64 64-64-64 64 64-64-64 64 64-64-64 64 64-64-64 64 64-64-64 64}
{61-73-46 82 31-88-13 90 -4-90 22 85-38-78 54 67-67-54 78 38-85-22 90 4-90 13 88-31-82 46 73-61}
{57-80-25 90 -9-87 43 70-70-43 87 9-90 25 80-57-57 80 25-90 9 87-43-70 70 43-87 -9 90-25-80 57}
{54-85 -4 88-46-61 82 13-90 38 67-78-22 90-31-73 73 31-90 22 78-67-38 90-13-82 61 46-88 4 85-54}
{50-89 18 75-75-18 89-50-50 89-18-75 75 18-89 50 50-89 18 75-75-18 89-50-50 89-18-75 75 18-89 50}

{46-90 38 54-90 31 61-88 22 67-85 13 73-82 4 78-78 -4 82-73-13 85-67-22 88-61-31 90-54-38 90-46}
{43-90 57 25-87 70 9-80 80 -9-70 87-25-57 90-43-43 90-57-25 87-70 -9 80-80 9 70-87 25 57-90 43}
{38-88 73 -4-67 90-46-31 85-78 13 61-90 54 22-82 82-22-54 90-61-13 78-85 31 46-90 67 4-73 88-38}
{36-83 83-36-36 83-83 36 36-83 83-36-36 83-83 36 36-83 83-36-36 83-83 36 36-83 83-36-36 83-83 36}
{31-78 90-61 4 54-88 82-38-22 73-90 67-13-46 85-85 46 13-67 90-73 22 38-82 88-54 -4 61-90 78-31}
{25-70 90-80 43 9-57 87-87 57 -9-43 80-90 70-25-25 70-90 80-43 -9 57-87 87-57 9 43-80 90-70 25}
{22-61 85-90 73-38 -4 46-78 90-82 54-13-31 67-88 88-67 31 13-54 82-90 78-46 4 38-73 90-85 61-22}
{18-50 75-89 89-75 50-18-18 50-75 89-89 75-50 18 18-50 75-89 89-75 50-18-18 50-75 89-89 75-50 18}
{13-38 61-78 88-90 85-73 54-31 4 22-46 67-82 90-90 82-67 46-22 -4 31-54 73-85 90-88 78-61 38-13}
{ 9-25 43-57 70-80 87-90 90-87 80-70 57-43 25 -9 -9 25-43 57-70 80-87 90-90 87-80 70-57 43-25 9}
{ 4-13 22-31 38-46 54-61 67-73 78-82 85-88 90-90 90-90 88-85 82-78 73-67 61-54 46-38 31-22 13 -4}

Two-dimensional transforms are computed by applying 1-D transforms in the horizontal and vertical directions. The elements of the core transform matrices were derived by approximating scaled DCT basis functions. For the transform block size of 4×4, an alternative integer transform derived from a DST is applied to the luma residual blocks for intrapicture prediction modes.

Since the rows of the transform matrix are close approximations of values of uniformly scaled basis functions of the orthonormal DCT, the prescaling operation that is incorporated in the dequantization of H.264/MPEG-4 AVC is not needed in HEVC. For quantization, HEVC uses essentially the same uniform reconstruction quantizer (URQ) scheme controlled by a quantization parameter (QP) as in H.264/MPEG-4 AVC. The range of the QP values is defined from 0 to 51, and an increase by 6 doubles the quantization step size such that the mapping of QP values to step sizes is approximately logarithmic. Quantization scaling matrices are also supported.

*10. Entropy coding:* Context adaptive binary arithmetic coding (CABAC) is used for entropy coding [1]. This is similar to the CABAC scheme in H.264/MPEG-4 AVC [6], but has undergone several improvements to improve its throughput speed (especially for parallel-processing architectures) and its compression performance, and to reduce its context memory requirements.

*11. In-loop deblocking filtering*: A deblocking filter similar to the one used in H.264/MPEG-4 AVC is operated within the interpicture prediction loop. However, the design is simplified in regard to its decision-making and filtering processes, and is made more friendly to parallel processing.

*12. Sample adaptive offset (SAO)*: Sample adaptive offset is applied to the reconstruction signal after the deblocking filter by using the offsets given in each CTB [14]. The encoder makes a decision on whether or not the SAO is applied for current slice. If SAO is enabled for current slice, the current slice allows each CTB select one of five SAO types as shown in Table III. The concept of SAO is to classify pixels into categories and reduces the distortion by adding an offset to pixels of each category. SAO operation includes Edge Offset (EO) which uses edge properties for pixel classification as SAO type 1-4 and Band Offset (BO) which uses pixel intensity for pixel classification as SAO type 5. Each CTB will have its own SAO parameters include sao_merge_left_flag, sao_merge_up_flag, SAO type and four offsets. If sao_merge_left_flag is equal to 1 current CTB will reuse the SAO type and offsets of left CTB, otherwise current CTB will not reuse SAO type and offsets of left CTB. If sao_merge_up_flag is equal to 1, current CTB will reuse SAO type and offsets of upper CTB, otherwise current CTB will not reuse SAO type and offsets of upper CTB.

**TABLE III**
**SPECIFICATION OF SAO TYPE [14]**

| SAO type | Sample adaptive offset type to be used | Number of categories |
|---|---|---|
| 0 | None | 0 |
| 1 | 1-D 0-degree pattern edge offset | 4 |
| 2 | 1-D 90-degree pattern edge offset | 4 |
| 3 | 1-D 135-degree pattern edge offset | 4 |
| 4 | 1-D 45-degree pattern edge offset | 4 |
| 5 | band offset | 4 |

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

*13. Special "Transform Skip" Coding Modes:* For certain types of content (especially screen content with graphics and text elements) more efficient compression is sometimes achieved when the transform is skipped (i.e. the residual is directly quantized and entropy coded) [2]. Furthermore, it is also possible to skip the quantization and loop filtering processes to enable lossless encoding of CUs.

### C. Encoder and Decoder in HEVC

A number of design aspects new to the HEVC standard improve flexibility for operation over a variety of applications and network environments and improve robustness to data losses [1]. However, the high-level syntax architecture used in the H.264/MPEG-4 AVC standard [6] has generally been retained, including the following features.

*1. Parameter set structure*: Parameter sets contain information that can be shared for the decoding of several regions of the decoded video [1]. The parameter set structure provides a robust mechanism for conveying data that are essential to the decoding process. The concepts of sequence and picture parameter sets from H.264/MPEG-4 AVC are augmented by a new video parameter set (VPS) structure.

*2. NAL unit syntax structure*: Each syntax structure is placed into a logical data packet called a network abstraction layer (NAL) unit [1]. Using the content of a two-byte NAL unit header, it is possible to readily identify the purpose of the associated payload data.

*3. Slices*: A slice is a data structure that can be decoded independently from other slices of the same picture, in terms of entropy coding, signal prediction, and residual signal reconstruction [1]. A slice can either be an entire picture or a region of a picture. One of the main purposes of slices is resynchronization in the event of data losses. In the case of packetized transmission, the maximum number of payload bits within a slice is typically restricted, and the number of CTUs in the slice is often varied to minimize the packetization overhead while keeping the size of each packet within this bound.

*4. Supplemental enhancement information (SEI) and video usability information (VUI) metadata*: The syntax includes support for various types of metadata known as SEI and VUI [4] [1]. Such data provide information about the timing of the video pictures, the proper interpretation of the color space used in the video signal, 3-D stereoscopic frame packing information, other display hint information, and so on.

### D. Parallel Decoding Syntax and Modified Slice Structuring

Finally, four new features are introduced in the HEVC standard to enhance the parallel processing capability or modify the structuring of slice data for packetization purposes [1]. Each of them may have benefits in particular application contexts, and it is generally up to the implementer of an encoder or decoder to determine whether and how to take advantage of these features.

*1. Tiles*: The option to partition a picture into rectangular regions called tiles has been specified [1]. The main purpose of tiles is to increase the capability for parallel processing rather than provide error resilience. Tiles are independently decodable regions of a picture that are encoded with some shared header information. Tiles can additionally be used for the purpose of spatial random access to local regions of video pictures. A typical tile configuration of a picture consists of segmenting the picture into rectangular regions with approximately equal numbers of CTUs in each tile. Tiles provide parallelism at a more coarse level of granularity (picture/ subpicture), and no sophisticated synchronization of threads is necessary for their use.

*2. Wavefront parallel processing*: When wavefront parallel processing (WPP) is enabled, a slice is divided into rows of CTUs [1] [14]. The first row is processed in an ordinary way, the second row can begin to be processed after only two CTUs have been processed in the first row, and the third row can begin to be processed after only two CTUs have been processed in the second row, and so on. The context models of the entropy coder in each row are inferred from those in the preceding row with a two-CTU processing lag. WPP provides a form of processing parallelism at a rather fine level of granularity, i.e., within a slice. WPP may often provide better compression performance than tiles (and avoid some visual artifacts that may be induced by using tiles). Fig. XIV (C) illustrates wavefront parallel processing.

*3. Dependent slice segments*: A structure called a dependent slice segment allows data associated with a particular wavefront entry point or tile to be carried in a separate NAL unit [1], and thus potentially makes that data available to a system for fragmented packetization with lower latency than if it were all coded together in one slice. A dependent slice segment for a wavefront entry point can only be decoded after at least part of the decoding process of another slice segment has been performed.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

Dependent slice segments are mainly useful in low-delay encoding, where other parallel tools might penalize compression performance.

Coding tools between H.264/MPEG-4 AVC and HEVC are compared in Table IV [63].

**TABLE IV**
**COMPARISON OF CODING TOOLS BETWEEN H.264/MPEG-4 AVC AND HEVC [63]**

| Algorithmic Element | H.264/MPEG-4 AVC | HEVC |
|---|---|---|
| Block structure | • MB | • CU<br>• PU<br>• TU |
| Inter prediction | • Spatial motion vector prediction (median)<br>• Direct mode | • AMVP<br>• Merge mode |
| Interpolation | • 6-tap FIR filter<br>• Bi-linear interpolation | • DCT-based interpolation (7-tap, 8-tap filter) |
| Intra prediction | • 9-modes (4×4, 8×8 luma blocks)<br>• 4-modes (16×16 luma and chroma blocks) | • 35 modes (planar, dc, 33 angular modes) |
| In-loop filtering | • Deblocking filter | • Simplified deblocking filter |
| Entropy coding | • CABAC<br>• CAVLC | • Simplified CABAC |

## III. COMPLEXITY ANALYSIS OF INTER PREDICTION

To analyze the complexity of inter prediction, authors in [63] used the sequences of Class A and Class B under the main profile–random access (MP–RA) configuration recommended by JCT-VC. As shown in Figs. XVIII and XIX, inter prediction has the most computational complexity, which consumes about 70% of total encoding time (TET).

In particular, ME of inter prediction accounts for the most complexity because SKIP/MERGE prediction omits the ME process and obtains the motion information, including MV, reference index, and IPM, from spatio-temporal neighboring PUs.

They [63] evaluated the computational complexity for each inter prediction mode (IPM). Figure XX shows that uniprediction has higher complexity than that of Bi. Also, the complexity of Uni-L0 is higher than that of Uni-L1 between unipredictions. It means that if the reference frame of Uni-L1 is the same as that of Uni-L0, then the MV of Uni-L1 is only copied from Uni-L0 to avoid redundant ME on the same reference frame. The reason why Bi is less complex than unipredictions is because it reuses the MV information from Uni-L0 or Uni-L1 without the AMVP process and because the integer ME is simply performed with a small search range of four.

According to complexity analysis, the inter prediction process imposes a heavy complexity burden of up to 70% on the entire encoding process. Regardless of the complexity distribution of IPMs, Bi is more likely to be selected than Uni-L0 or Uni-L1, as shown in Fig. XXI. Therefore, if accurate IPM is predetermined among Uni-L0, Uni-L1, and Bi, then the ME complexity incurred by unnecessary IPM can be significantly removed [63].

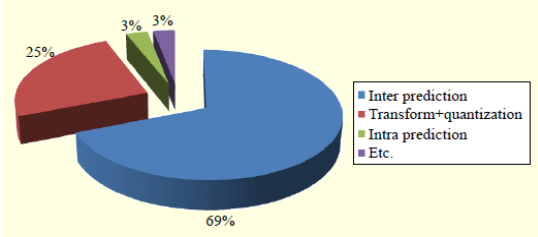



**FIGURE XVIII COMPLEXITY DISTRIBUTION OF TOTAL ENCODING TIME [63]**

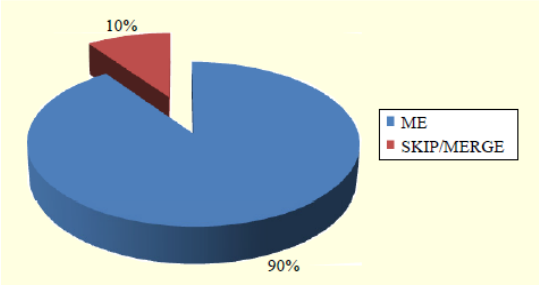



**FIGURE XIX COMPLEXITY DISTRIBUTION OF INTER PREDICTION [63]**

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

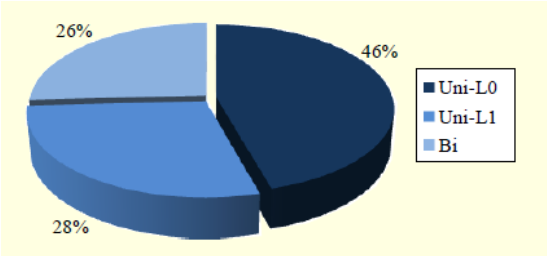



**FIGURE XX COMPLEXITY DISTRIBUTION ACCORDING TO INTER PREDICTION [63]**

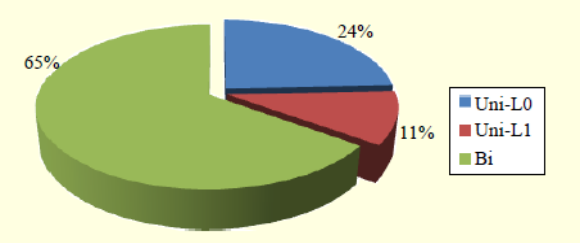



**FIGURE XXI DISTRIBUTION OF BEST IPM [63]**

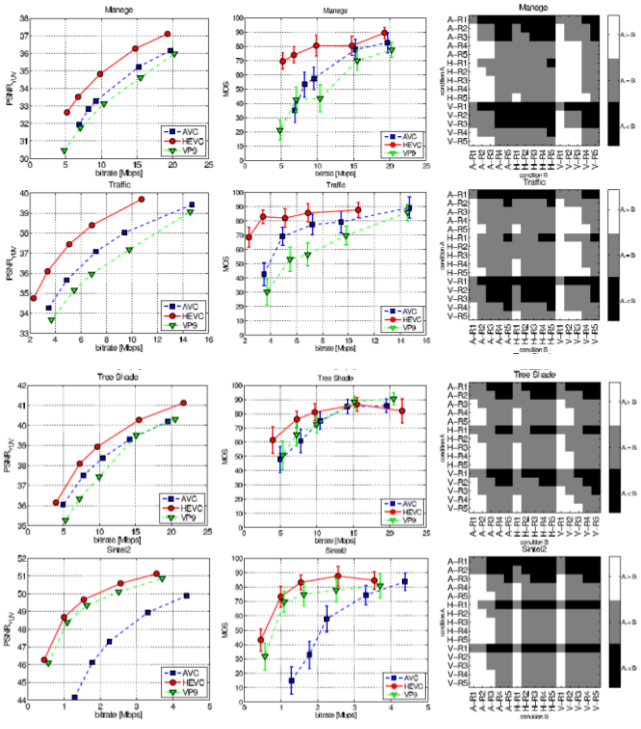


**FIGURE XXII PSNR RD GRAPHS (LEFT COLUMN) AND SUBJECTIVE RATINGS SHOWING MEAN OPINION SCORES AND CONFIDENCE INTERVALS (MIDDLE COLUMN) FOR DIFFERENT BIT-RATE AND CONTENT SEPARATELY. RIGHT COLUMN PRESENTS A MULTIPLE COMPARISON TEST FOR ALL POSSIBLE COMBINATIONS OF CODECS [59]**

In the graph above all possible combinations of codecs are - A stands for AVC, H stands for HEVC, and V stands for VP9 coding algorithm. Bit-rates are from R1 to R5 for which each test content is tested separately. In each plot, the color of each square shows the result of the significance test between the mean opinion scores related to the two test conditions reported in the corresponding column and row. A white (black) square indicates that the MOS corresponding to condition A is statistically significantly better (worse) than the MOS corresponding to condition B while a grey square indicates that the two MOSs are statistically not different.

**TABLE V**
**Comparison of investigated coding algorithms in terms of bit-rate reduction for similar PSNRYUV and MOS. Negative values indicate actual bit-rate reduction [59]**

| SL No | HEVC vs. AVC | | VP9 vs. AVC | | HEVC vs. VP9 | |
|---|---|---|---|---|---|---|
| | **BD-PSNR** | **BD-MOS** | **BD-PSNR** | **BD-MOS** | **BD-PSNR** | **BD-MOS** |
| 1. | -28.7% | -44.6% | 10.6% | 29.2% | -39.7% | -63.7% |
| 2. | -37.2% | -57.5% | 25.1% | 61.0% | -49.8% | -74.7% |
| 3. | -22.7% | -37.4% | 18.9% | -8.2% | -33.7% | -31.9% |
| 4. | -69.9% | -70.9% | -60.9% | -61.7% | -19.0% | -27.5% |
| **Avg** | **-39.6%** | **-52.6%** | **-1.59%** | **5.1%** | **-35.6%** | **-49.4%** |

Contents are SL No1. Manege, 2. Traffic, 3. Tree shade, 4. Sintel2

Several techniques [65, 66, 67] for detection of zero blocks in HEVC have been investigated with a view to reduce the encoder complexity. These can lead to several research topics.

## IV. HEVC LOSSLESS CODING

Video content containing computer generated objects is usually denoted as screen content [49] [50] and is becoming popular in applications such as desktop sharing, wireless displays, etc. Screen content images and videos are characterized by high frequency details such as shar edges and high contrast image areas.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

On these areas classical lossy encoding tools – spatial transform plus quantization – may significantly compromise their quality and intelligibility. Therefore, lossless coding is used instead and improved coding tools should be specifically devised for screen content.

Some companies actively working on Screen Content Coding (SCC) [51] are MediaTek, Qualcomm, Microsoft, Interdigital, Huawei etc.

Some practical reasons [52] for lossless compression are:

1. Medical diagnosis, fine art archive, military purposes, surveillance etc.
2. Image/video analysis, automatic product examination. The observer is not a human — "perceptually lossless" does not make sense.
3. Contribution transmission, studio applications. Subject to be re-encoded/edited repeatedly — coding error accumulates, if lossy.

Some emotional reasons [52] are:

1. A CG artist said, "Just do not lose even one bit from my artwork!"
2. Provides an upper-bound of image quality (can feel at ease).
3. Fed up with drawing R-D graphs [53], evaluating Lagrange cost function, subjective impairment measurement, HVS.

The simple lossless coding mode is to bypass quantization and inverse quantization as it was used in H.264/AVC [18]. Figure XXIII [54] [55] illustrates the HEVC encoder block diagram with quantization and inverse quantization bypassed. In the lossless mode, the de-blocking filter [56] and SAO [57] are also disabled. This lossless mode serves as the lossless anchor methods used in this contribution. Figure XXIII [54] [55] shows the HEVC encoder with lossless coding mode that bypasses transform, quantization, and disables de-blocking, SAO and ALF.

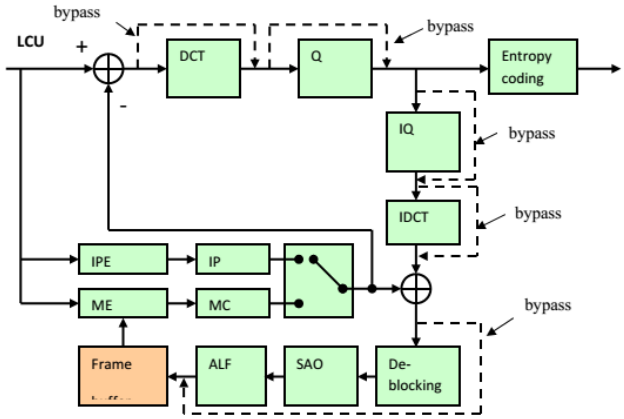



**FIGURE XXIII HEVC encoder with lossless coding mode that bypasses transform, quantization, and disables de-blocking, SAO and ALF [54] [55]**

The major consideration for the screen content is that only lossless coding should be used which has the compression ratio of 2 or 3 instead of compression ratio of 10 to 1000 for lossy compression [58]. The sample-based angular intra prediction (SAP) [54] [55] is designed to better exploit the spatial redundancy in the lossless coding mode by generating intra prediction samples from adjacent neighbors. Now the SAP can be used to implement the lossless coding for the screen content in HEVC by exploiting the spatial redundancy so that the finer details of the screen content can be intact.

## V. Lossy To Lossless Scalable Video Coding System Based On HEVC

In this section, the corresponding block diagrams of scalable lossless HEVC video coding, results (graphs) and comparison with JPEG-LS [70] are introduced. Lossless compression of image and video content is desired in many professional applications like medical imaging, surveillance systems, and archiving systems.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

However, a lossless representation usually requires much more storage capacity compared to lossy compression and there are scenarios where it is not possible to deal with the huge amount of data for lossless coding. In these cases, a lossy version of the data for preview purposes would be favorable. Afterwards, in a second step, the lossy preview may be refined to lossless quality, at least for certain selected parts of the coded video. In order to fulfill all these demands Heindel et al [62] investigate a very flexible and low-complex scalable video coding scheme using a lossy base layer (BL) compressed with HEVC. The considered system has already been introduced in Ref. [61]. In [62] different spatial prediction algorithms for lossless coding of the enhancement layer (EL) are examined.

Starting with the lossy coded BL, residual images are computed by subtracting the reconstructed pictures from the original ones. Subsequently, each residual image is losslessly compressed by means of spatial prediction and entropy coding. For entropy coding they employ Context-based Adaptive Binary Arithmetic Coding (CABAC), which is also included in the HEVC framework. Intra prediction using Sample-based Weighted Prediction (SWP) has already delivered good compression results in [61]. Consequently, they named the complete scalable system SELC (SWP for Enhancement Layer Coding). SWP implicitly performs denoising during prediction and has therefore been selected as favorable prediction scheme for residual images, which reveal a considerable amount of noise. However, it turned out that SWP is not always the optimal prediction scheme. Therefore, in [62] they compare the compression performance of SWP to other well-established prediction methods, namely Edge Directed Prediction (EDP) and the Median Edge Detector (MED). Using EDP, the prediction coefficients are determined using autoregressive least squares optimization. MED, also known as LOCO-I algorithm, is a very simple non-linear predictor. Due to the dominating noise part within the residual images, the performance of the examined algorithms cannot be deduced from other studies based on usual natural images.

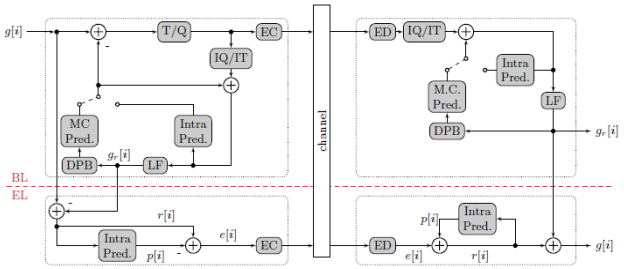



**FIGURE XXIV Detailed structure of the proposed system allowing lossy to lossless scalable video transmission [62]**

Lossy to lossless scalable transmission is realized by employing a coding system as depicted in Fig. XXIV. For compression of the lossy BL a coder compliant with the HEVC standard is used. All parts of the scalable system belonging to the BL encoder and decoder are shown in the upper part of Fig. XXIV. The individual blocks for BL coding denote the following components:

- transformation and quantization (T/Q),
- inverse quantization and inverse transformation (IQ/IT),
- loop filters (LF),
- decoded picture buffer (DPB),
- motion compensated prediction (MC Pred.),
- entropy coding (EC), and entropy decoding (ED).

The original input signal entering the coding system is denoted as $g[\boldsymbol{i}]$, where the two-dimensional index $\boldsymbol{i}$ represents the spatial coordinates of the individual samples within the image. The lossy reconstructed output is referred to as $g_r[\boldsymbol{i}]$. Note that $g_r[\boldsymbol{i}]$ is not only available at the output of the BL decoder, but also accessible within the decoding loop of the BL encoder.

Besides that, the encoder as well as the decoder for the EL are shown in the lower part of Fig. XXIV. The signal which has to be encoded in the EL is computed first as the difference between $g[\boldsymbol{i}]$ and $g_r[\boldsymbol{i}]$. The resulting signal is called “residual" and is denoted as $r[\boldsymbol{i}]$, i.e.,

$$r[\boldsymbol{i}] = g[\boldsymbol{i}] - g_r[\boldsymbol{i}] \quad (1)$$

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

Intra prediction (Intra Pred.) is performed in order to exploit the remaining spatial correlation within $r[\boldsymbol{i}]$. Finally, the obtained prediction error $e[\boldsymbol{i}]$ is entropy coded (EC), resulting in the EL bitstream.

*Adaptive prediction decision:* Spatial prediction of $r[\boldsymbol{i}]$ may lead to an increased bitrate of the EL bitstream compared to directly encoding $r[\boldsymbol{i}]$ in some cases. In general, choosing a low value for the quantization parameter (QP) of the BL encoder makes spatial prediction of $r[\boldsymbol{i}]$ very difficult, because $r[\boldsymbol{i}]$ consists mainly of noise-like components in this case. However, the QP threshold where prediction starts to fail is not clearly recognizable.

It is checked whether the sum of absolute values of the predicted signal is reduced compared to the sum of absolute values of the signal before prediction. If this is true, the prediction step is enabled. Otherwise prediction is skipped and $e[\boldsymbol{i}]$ is set to be equal to $r[\boldsymbol{i}]$. This decision, whether to enable prediction or not, is made once for the luminance channel and once for both chrominance channels of a picture. Finally, one bit for each of the decisions is used for explicit signaling in the bitstream.

### *A. Sample-based Weighted Prediction (SWP)*

The SWP algorithm has been introduced in [61] for improving the efficiency of lossless coding using the HEVC standard. Its basic idea is to predict the value of the current sample as a weighted sum of four neighboring sample values. The used neighboring samples originate from a causal neighborhood, which is depicted in Fig. XXV(a), where *X* is the current sample to be predicted and *a*, *b*, *c*, and *d* are the samples in the neighborhood.

In order to assign weights to these four samples of the neighborhood, a so-called patch is introduced. This patch is used to estimate similarities between the current sample and the samples from the neighborhood. The patch has the same shape as the neighborhood and each sample of the neighborhood is surrounded by its own patch. Patches corresponding to the samples *X* and *b* are shown in Fig. XXV(b) and Fig. XXV(c), respectively. The similarity between *X* and a sample from the neighborhood is expressed as the sum of absolute differences (SAD) between the samples in the corresponding patches. Finally, the obtained SAD values are mapped to weights, which are used to calculate the prediction as a weighted sum of the neighborhood sample values.

Formally, the generation of the prediction value $p_{\mathrm{SWP}}[i]$ of the current residual sample $r[i]$ can be expressed as

$$p_{\mathrm{SWP}}[\boldsymbol{i}] = \left\lfloor \frac{\sum_{\boldsymbol{j}\in\mathbf{S}} w_{\boldsymbol{i},\boldsymbol{j}} \cdot r[\boldsymbol{j}] + (\sum_{\boldsymbol{j}\in\mathbf{S}} w_{\boldsymbol{i},\boldsymbol{j}})/2}{\sum_{\boldsymbol{j}\in\mathbf{S}} w_{\boldsymbol{i},\boldsymbol{j}}} \right\rfloor, \tag{2}$$

where $\boldsymbol{i}$ and $\boldsymbol{j}$ are the two-dimensional pixel coordinates, $S$ denotes the set of coordinates in the neighborhood, and $w_{i,j}$ are the integer weights for the individual samples. The operator $\lfloor . \rfloor$ indicates rounding towards minus infinity. In order to obtain the weights, the SAD values

$$\mathrm{SAD}_{\boldsymbol{i},\boldsymbol{j}} = \sum_{\mathbf{n}\in\mathbf{N}_0} |r[\boldsymbol{i}+\mathbf{n}] - r[\boldsymbol{j}+\mathbf{n}]| \tag{3}$$

between the patch of the current sample $r[\boldsymbol{i}]$ and the patches of the neighborhood samples $r[\boldsymbol{j}]$ are calculated. In (3), $\mathbf{n}$ denotes the relative shift from the actual sample position to the particular neighborhood pixel and $\mathbf{N}_0$ is the set of all shifts, i.e.,

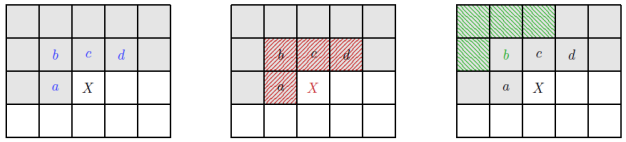
$$\mathbf{N}_0 \in \{(-1,0),(-1,-1),(0,-1),(1,-1)\}. \tag{4}$$

Finally, the obtained SAD values are mapped to the weights $w_{i,j}$ by the exponentially decaying function

$$\mathrm{LUT}[\mathrm{SAD}] = \mathrm{round}\left(a \cdot 2^{-\mathrm{SAD}/h}\right), \tag{5}$$

where the possible function values are precomputed and stored in a look-up table (LUT), which retains the complexity low, i.e.,

$$w_{\boldsymbol{i},\boldsymbol{j}} = \mathrm{LUT}[\mathrm{SAD}_{\boldsymbol{i},\boldsymbol{j}}]. \tag{6}$$




**(a)** **(b)** **(c)**

**FIGURE XXV** (A) **Current sample X with its neighborhood a, b, c, and d;** (B) **a patch of X with a red hatch pattern;** (C) **a patch of b with a green hatch pattern. [62]**

### *B. Median Edge Detector (MED) Prediction*

The LOCO-I algorithm for efficient and low-complex spatial prediction for lossless coding of images was finally standardized into JPEG-LS. Due to its alternative interpretation of choosing the median of three predictors, it was renamed to Median Edge Detector (MED) during the standardization process. Thus, the algorithm is referred to as MED in the remainder of this section.

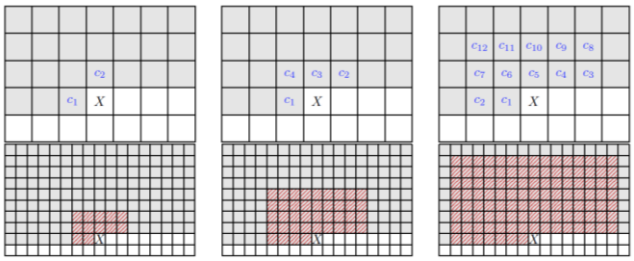
The MED algorithm provides a prediction for the current pixel based on a small neighborhood consisting of three samples only. Referring to Fig. XXV(a), those are the samples *a*, *b*, and *c*. The predicted value is obtained from the equation

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

$$p_{\mathrm{MED}}[\boldsymbol{i}] = \begin{cases} \min(a_{\boldsymbol{i}}, c_{\boldsymbol{i}}) & \text{if } b_{\boldsymbol{i}} > \max(a_{\boldsymbol{i}}, c_{\boldsymbol{i}}) \\ \max(a_{\boldsymbol{i}}, c_{\boldsymbol{i}}) & \text{if } b_{\boldsymbol{i}} < \min(a_{\boldsymbol{i}}, c_{\boldsymbol{i}}) \\ a_{\boldsymbol{i}} + c_{\boldsymbol{i}} - b_{\boldsymbol{i}} & \text{else,} \end{cases} \tag{7}$$

where $a_i = r[\boldsymbol{i} + (-1,0)]$, $b_i = r[\boldsymbol{i} + (-1,-1)]$, and $c_i = r[\boldsymbol{i} + (0,-1)]$.

By the application of these prediction rules, directional edges within the image are taken into account. Prediction along horizontal or vertical edges is implicitly handled by the first and the second condition in (7). If no edge is recognized, the third row of (7) is employed, which yields a planar prediction assuming the current sample lies on a plane defined by the sample values of *a*, *b*, and *c*.




(a) $N = 2$, $T = 2$ (b) $N = 4$, $T = 4$ (c) $N = 12$, $T = 7$

**FIGURE XXVI CONTEXTS (TOP ROW) AND TRAINING WINDOWS (BOTTOM ROW) FOR EDP USING DIFFERENT PREDICTION ORDERS N [62]**

### *C. Edge-Directed Prediction (EDP)*

Edge-Directed Prediction (EDP) is a spatial prediction algorithm based on Least Squares (LS) optimization producing a prediction of the current pixel based on a causal neighborhood referred to as "context". Similar to the SWP algorithm the predicted value is obtained as a weighted sum of the context samples. However, the weights are determined in a different way,

i.e., by using autoregressive LS optimization incorporating an additional "training window".

In the paper [62] the training window is chosen as suggested in [64]. A causal neighborhood ranging *T* pixels to the left, top and right of the current pixel is used, where $T = \min(N,7)$ and *N* is the prediction order, i.e., the number of pixels in the context. *T* is limited to 7 like in [64], where it is claimed that a larger window size does not further improve the prediction performance.

For the evaluation authors on [62] choose the three different prediction orders $N = 2$, $N = 4$, and $N = 12$. The corresponding contexts and training windows are depicted in Fig. XXVI.

Incorporating the samples of the context, the prediction is calculated as the weighted sum

$$p_{\mathrm{EDP}}[\boldsymbol{i}] = \sum_{\boldsymbol{j} \in \mathbf{S}} w_{\boldsymbol{i},\boldsymbol{j}} \cdot r[\boldsymbol{j}], \tag{8}$$

where $w_{i,j}$ is the weight for the context sample $r[\boldsymbol{j}]$ belonging to the context of the pixel $r[\boldsymbol{i}]$. Again, analogously to (2), **S** is the set of all pixel coordinates in the context (i.e., the coordinates of $c_1$ to $c_N$ in Fig. XXVI).

Sticking to the notation of [64], the training window contains $M = 2T(T + 1)$ pixels which are combined to the vector $\vec{\boldsymbol{y}}_{\boldsymbol{i}}$ of length *M*. Each entry $y_{i,m}$ ($m = 1, \ldots, M$) of $y_i$ has its own context used for prediction of the particular sample. Writing the sample values of the context of $y_{i,m}$ into the m-th row of a matrix $\mathbf{C}_{\boldsymbol{i}}$ for all *m*, we obtain an $M \times N$ matrix $\mathbf{C}_{\boldsymbol{i}}$ which is used to optimize the prediction weights $w_{i,j}$. Combining also the weights to a vector $\vec{\boldsymbol{w}}_{\boldsymbol{i}}$ of length *N*, the prediction process of all training samples of $r[\boldsymbol{i}]$ can be written as

$$\vec{\boldsymbol{y}}_{\boldsymbol{i}} = \mathbf{C}_{\boldsymbol{i}} \vec{\boldsymbol{w}}_{\boldsymbol{i}}. \tag{9}$$

In order to obtain optimal weights, the LS optimization problem

$$\min \| \vec{\boldsymbol{y}}_{\boldsymbol{i}} - \mathbf{C}_{\boldsymbol{i}} \vec{\boldsymbol{w}}_{\boldsymbol{i}} \|_2 \tag{10}$$

is solved, which results in the closed-form expression

$$\vec{\boldsymbol{w}}_{\boldsymbol{i}} = (\mathbf{C}_{\boldsymbol{i}}^T \mathbf{C}_{\boldsymbol{i}})^{-1} (\mathbf{C}_{\boldsymbol{i}}^T \vec{\boldsymbol{y}}_{\boldsymbol{i}}). \tag{11}$$

It should be noted that the presented update of the prediction weights is performed for each single sample of the picture, i.e., Equation (11) is solved for every pixel. The irregular update of the weights as proposed in [64] which reduces the computational complexity, but also the prediction efficiency, is not applied.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

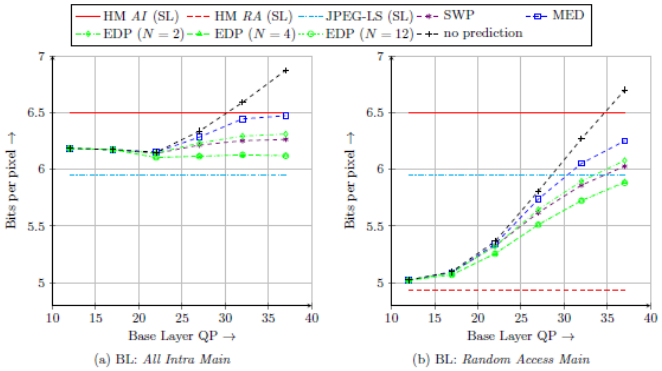



**FIGURE XXVII BITS PER PIXEL FOR LOSSLESS SINGLE-LAYER (SL) CODING WITH HM-11.0 AND JPEG-LS COMPARED TO THE LOSSY TO LOSSLESS SCHEME WITH DIFFERENT PREDICTION ALGORITHMS FOR EL CODING. AVERAGED RESULTS FOR CLASS C SEQUENCES. HM-11.0 SL CODING USING ALL INTRA MAIN (AI) AND RANDOM ACCESS MAIN (RA) CONFIGURATION. BL COMPRESSION USING ALL INTRA MAIN (A) RANDOM ACCESS MAIN (B) CONFIGURATION SETTINGS [62]**

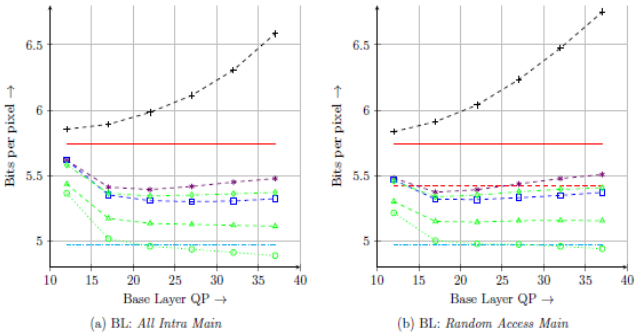



**FIGURE XXVIII BITS PER PIXEL FOR LOSSLESS SINGLE-LAYER (SL) CODING WITH HM-11.0 AND JPEG-LS COMPARED TO THE LOSSY TO LOSSLESS SCHEME WITH DIFFERENT PREDICTION ALGORITHMS FOR EL CODING. AVERAGED RESULTS FOR CT SEQUENCES. HM-11.0 SL CODING USING ALL INTRA MAIN (AI) AND RANDOM ACCESS MAIN (RA) CONFIGURATION. BL COMPRESSION USING ALL INTRA MAIN (A) RANDOM ACCESS MAIN (B) CONFIGURATION SETTINGS [62]**

In [62] a system layout for scalable lossless video coding using two layers has been proposed. The system employs an HEVC compliant encoder for the lossy BL, while the EL is coded using spatial prediction of the residual signal followed by entropy coding of the prediction error. Several algorithms have been evaluated for the task of intra prediction of the residual signal. It turned out that SWP is a better choice than MED for usual consumer video sequences recorded by normal video cameras.

Nevertheless, the coding efficiency for the EL may be improved by investing more processing time, i.e., by employing the EDP algorithm, where a prediction order of 4 is already sufficient. In doing so, in total only about 2.7% additional data rate is needed for lossy to lossless scalable coding compared to SL JPEG-LS compression. Considering *CT data*, MED outperforms SWP. Furthermore, also for this type of image content EDP yields additional gains, where a prediction order of $N$ = 12 is advantageous to $N$ = 4. With $N$ = 12, scalable coding of the *CT sequences* is even more efficient than SL JPEG-LS coding for certain values of QP. To summarize, the used prediction algorithm should be chosen depending on the image content to code, which is usually known beforehand. Furthermore, low-complex implementations of the EDP algorithm may also be considered, as EDP is able to improve the efficiency considerably.

In future work the efficiency of the scalable coding system shall be further increased by processing the residual images in a block-wise manner. This allows a locally adaptive and thus more accurate decision to enable or disable prediction. Moreover, the necessity to update the Rice parameter $k$ very frequently is reduced by this approach [62].

## VI. COMPARISON OF HEVC VS. VP9 AND X264 FOR LOW DELAY VIDEO APPLICATIONS

A comparative assessment for the low delay configuration of HEVC, VP9, and H.264/MPEG-4 AVC encoders was presented by D. Grois et al in [60]. They used video conferencing sequences referred to as Class E for experiments. For evaluating H.264/MPEG-4 AVC, the open-source x264 High Profile encoder was selected.

As it is clearly seen from Table VI, the HEVC encoder provides significant gains in terms of coding efficiency compared to both VP9 and x264 encoders. Particularly, for FourPeople and KristenAndSara video sequences, the R-D curves of VP9 and x264 encoders are very close, while the calculated BD-BR savings of the x264 encoder vs. VP9 encoder for the FourPeople video sequence are 5.8% in favor of x264. [60]

Figure XXX presents R-D curves of HEVC, x264, and VP9 encoders, along with corresponding HEVC bit-rate savings for typical 2-pass encoding examples of the tested video sequences. [60]

As it is observed from Fig. XXX, similarly to the 1-pass encoding mode, the HEVC encoder provides significant gains in terms of coding efficiency compared to both VP9 and x264 encoders.

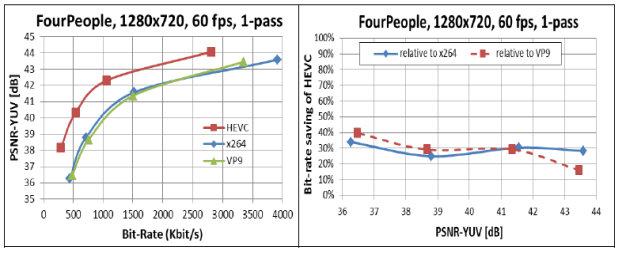
International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

In this case again, the Rate-Distortion (RD) curves of the FourPeople and KristenAndSara video sequences for the VP9 and x264 encoding are also very close, while the difference in BD-BR savings of the VP9 encoder vs. x264 encoder for the FourPeople video sequence are 2.5% in favor of VP9. [60]

According to the detailed experimental results, the coding efficiency of VP9 was shown to be inferior to HEVC with an average bit-rate overhead of 32.5% at the same objective quality for the 1-pass encoding, and 32.6% for the 2-pass encoding (Tables VI and IX). Also, it was shown that the VP9 encoding times are larger by a factor of about 2,000 compared to those of the x264 High Profile encoder for the 1-pass encoding, and about 400 for the 2-pass encoding (Tables VIII and XI), as a trade-off of bit-rate savings of 12.5% and 14.6% for the 1-pass and 2-pass encoding, respectively. When compared to the full-fledged HEVC reference software encoder implementation, the VP9 encoding times are lower, at average, by a factor of about 2.7 and 6.1 for the 1-pass and 2-pass encoding, respectively. In addition, it was observed that the HEVC encoder provides significant average bit-rate savings of more than 40% relative to the x264 High Profile encoder for both 1-pass and 2-pass encoding (Tables VI, VII, IX, and X). [60]

**TABLE VI**
**HEVC CALCULATED BD-BR SAVINGS (COMPARED TO VP9 AND X264 HIGH PROFILE ENCODERS, 1-PASS MODE) [60]**

| | *HEVC vs. VP9 (in %)* | *HEVC vs. x264 (in %)* |
|---|---|---|
| Sequences/QPs | BD-BR | BD-BR |
| FourPeople | -31.6% | -27.6% |
| Johnny | -33.6% | -51.3% |
| KristenAndSara | -32.4% | -43.4% |
| **Averages** | **-32.5%** | **-40.8%** |



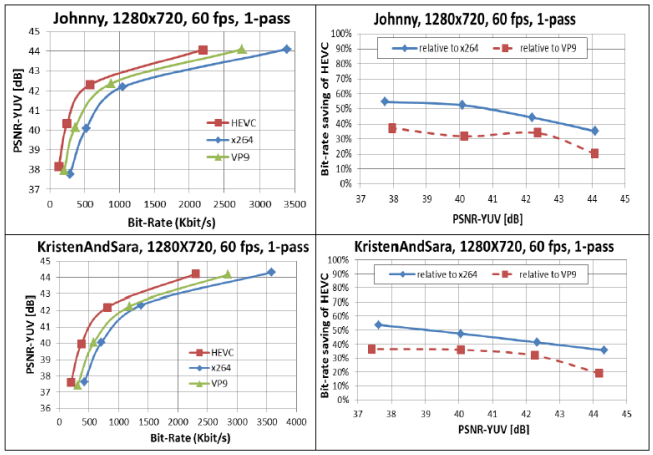




**FIGURE XXIX 1-PASS ENCODING MODE: R-D GRAPHS AND CORRESPONDING BIT-RATE SAVING PLOTS FOR THE TESTED CLASS E SEQUENCES, WHICH REFER TO DIFFERENT VIDEO CONFERENCING SCENARIOS. [60]**

**Table VII**
**Summarized BD-BR Low-Delay Experimental Results (Compared to VP9 and x264 High Profile Encoders, 1-Pass Mode) [60]**

| *CODEC* | HEVC | x264 | VP9 |
|---|---|---|---|
| HEVC | | -40.8% | -32.5% |
| x264 | 73.5% | | 17.0% |
| VP9 | 48.2% | -12.5% | |

**Table VIII**
**Encoding Low-Delay Run Times for Equal PSNRYUV (Compared to VP9 and x264 High Profile Encoders, 1-Pass Mode) [60]**

| | *HEVC vs. VP9 (in %)* | | | | *VP9 vs. x264 (in %)* | | | |
|---|---|---|---|---|---|---|---|---|
| Sequences/QPs | 22 | 27 | 32 | 37 | 22 | 27 | 32 | 37 |
| FourPeople | 325 | 274 | 253 | 239 | 160981 | 208502 | 266549 | 292080 |
| Johnny | 311 | 254 | 236 | 230 | 95691 | 120474 | 165065 | 235111 |
| KristenAndSara | 338 | 291 | 262 | 250 | 139382 | 172523 | 231339 | 291056 |
| Averages | 325 | 273 | 250 | 240 | 132018 | 167166 | 220984 | 272749 |

| Total Average | 272 | 198229 |
|---|---|---|

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

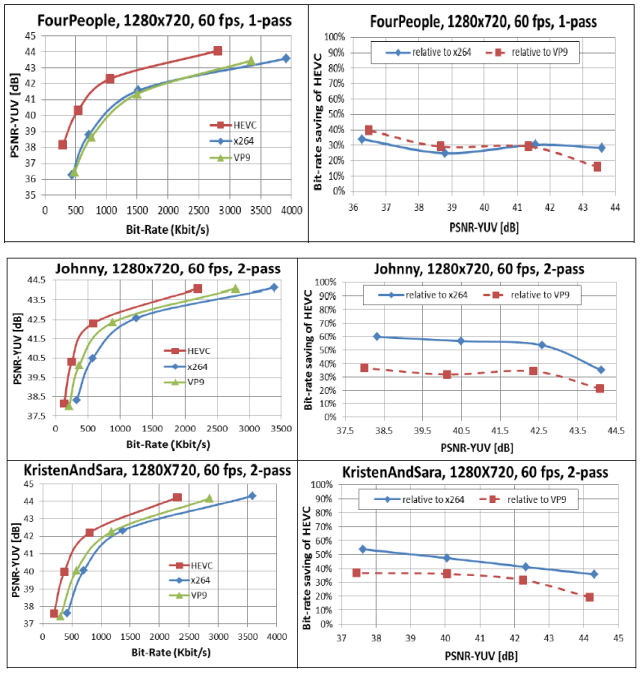



**FIGURE XXX 2-pass Encoding Mode: R-D graphs and corresponding bit-rate saving plots for the tested Class E sequences, which are representing different video conferencing scenarios [60]**

**Table IX**
**HEVC Bit-Rate Savings for Equal PSNRYUV (Compared to VP9 and x264 High Profile Encoders, 2-Pass Mode) [60]**

| | *HEVC vs. VP9 (in %)* | | | | | *HEVC vs. x264 (in %)* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Sequences/QPs | 22 | 27 | 32 | 37 | BD-BR | 22 | 27 | 32 | 37 | BD-BR |
| FourPeople | 16.8 | 28.7 | 33.5 | 39.6 | -31.1 | 30.1 | 34.2 | 31.2 | 40.3 | -32.7 |
| Johnny | 21.4 | 34.5 | 32.0 | 36.8 | -34.0 | 35.3 | 53.8 | 56.9 | 60.0 | -50.6 |
| KristenAndSara | 19.3 | 31.7 | 36.0 | 36.8 | -32.6 | 35.8 | 41.1 | 47.4 | 53.8 | -43.4 |
| Averages | 19.2 | 31.6 | 33.8 | 37.7 | -32.6 | 33.7 | 43.0 | 45.2 | 51.4 | -42.2 |
| Total Average | 30.6 | | | | -32.6 | 43.3 | | | | -42.2 |

**Table X**
**Summarized BD-BR Low-Delay Experimental Results (Compared to VP9 and x264 High Profile Encoders, 2-Pass Mode) [60]**

| *CODEC* | HEVC | x264 | VP9 |
|---|---|---|---|
| HEVC | | -42.2% | -32.6% |
| x264 | 75.8% | | 18.5% |
| VP9 | 48.3% | -14.6% | |

**Table XI**
**Encoding Low-Delay Run Times for Equal PSNRYUV (Compared to VP9 and x264 High Profile Encoders, 2-Pass Mode) [60]**

| | *HEVC vs. VP9 (in %)* | | | | *VP9 vs. x264 (in %)* | | | |
|---|---|---|---|---|---|---|---|---|
| Sequences/QPs | 22 | 27 | 32 | 37 | 22 | 27 | 32 | 37 |
| FourPeople | 637 | 569 | 554 | 544 | 32453 | 46370 | 57743 | 62669 |
| Johnny | 638 | 571 | 604 | 615 | 20660 | 25786 | 31495 | 41016 |
| KristenAndSara | 694 | 639 | 637 | 646 | 26128 | 34074 | 43020 | 51369 |
| Averages | 656 | 593 | 598 | 602 | 26414 | 35410 | 44086 | 51685 |
| Total Average | 612 | | | | 39399 | | | |

## VII. Extensions

While the first version of HEVC is sufficient to cover a wide range of applications, needs for enhancing the standard in several ways have been identified, including work on range extensions for color format and bit depth enhancement, embedded-bitstream scalability, and 3D video [2]. The standardization of extensions in each of these areas was completed in 2014.

### *A. Range Extension*

The range extensions for HEVC include support for the 4:2:2 and 4:4:4 enhanced chroma sampling structures and sample bit depths beyond 10 bits per sample [2]. Additional areas of work include coding of screen content (graphics and other noncamera- view or mixed content), very high bit-rate and lossless coding, coding of auxiliary pictures (e.g., alpha transparency planes), and direct coding of RGB source content. The range extensions were finalized in 2014, and the standard can be found in [3]. As previously mentioned, the 4:2:0 chroma format supported in the version 1 profiles has chroma information that is half resolution both in the horizontal and vertical dimensions. This has been typical for consumer entertainment use, but the demands of higher-quality applications and screen content coding require use of the 4:4:4 format with full-resolution chroma representations, or of the 4:2:2 format in which half-resolution horizontal but full-resolution vertical chroma sampling is used.

In the 4:4:4 case, the draft range extensions support two modes of operation. The first, known as separate color plane coding, is to process each of the three-color components separately, as if they were ordinary monochrome (luma) pictures. The second mode, known as joint color plane coding, is to process them jointly. Separate color plane coding is generally considered more difficult to support, so it is possible that this mode may not be supported in the final profile specifications.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

*B. Scalability Extensions*

The principle of scalable video encoder is to divide the traditional single stream video in a multi stream flow, composed by distinct and complementary components, often referred to as layers. Figure XXXI shows the concept of transmitter encoding the input video sequence into three complimentary layers [42]. The receiver can select and decode different number of layers each corresponding to distinct video characteristics in accordance with the processing constraints of both the network and device itself.

The usual modes of scalability are temporal, spatial, and quality scalability (Fig. XXXII). Spatial scalability and temporal scalability describe cases in which subsets of the bit stream represent the source content with a reduced picture size (spatial resolution) or frame rate (temporal resolution), respectively. With quality scalability, the sub-stream provides the same spatio-temporal resolution as the complete bit stream, but with a lower fidelity – where fidelity is often informally referred to as signal-to-noise ratio (SNR). Quality scalability is also commonly referred to as fidelity or SNR scalability [43].

The scalability extension to HEVC enables spatial and coarse grain SNR scalability, and is referred to as "SHVC" [2]. The finalization of this extension of HEVC was done by the end of 2014, and the draft text can be found in [4]. Temporal scalability support was already provided in HEVC version 1, and was combined with spatial and SNR scalability in SHVC.

Figure XXXIII shows the scalable video coding structure with three coding layers (spatial, temporal and SNR scalabilities). The SHVC design uses a multi-loop coding framework, such that in order to decode an enhancement layer, its reference layers have to first be fully decoded to make them available as prediction references.

This differs from AVC's SVC extension design, which used single-loop decoding for inter-coded macroblocks so that the motion compensation process would only need to be performed once when decoding. When two spatial or SNR layers are used, the base layer is the only reference layer, but for three or more spatial or SNR layers, intermediate layers may also be used as reference layers. To some extent, an efficient single-loop decoding was only possible by defining reference and enhancement layer decoding processes closely dependent at the block-level, e.g. adding new prediction modes, using reference layer contexts for the enhancement layer's entropy coding etc.

The high-level design of the HEVC scalability extension, e.g., multi-loop coder/decoder and restrictions against block-level changes, were motivated by ease of implementation, especially the possibility to re-use existing HEVC implementations, even though the overall number of computations and memory accesses of the decoder would be higher than in a single-loop design. Beyond that, multi-loop coding also provides coding efficiency advantages over single-loop coding designs. The coding tools in the HEVC scalability extension are limited to changes at the slice level and above. The reference layer picture, resampled if necessary, is used as additional reference picture for enhancement layer prediction, which enables interlayer texture and motion parameter prediction.

The multi-loop design is somewhat similar to AVC's and HEVC's multiview extensions, which require full decoding of the base view in the case of decoding dependent views. However, in the multiview case (as currently specified), all views have the same resolution so that no resampling is needed. The same applies for the case of SNR scalability, where scalable layers represent pictures of identical spatial resolution. The base layer bitstream can be interpreted by legacy decoders, and may be either an HEVC bitstream or an AVC bitstream. When the base layer is an AVC bitstream, only inter-layer texture prediction is performed, with inter-layer motion prediction not supported. Investigations have shown that the compression benefit would be small, and the AVC base layer motion vectors may not easily be accessible in existing decoder implementations.

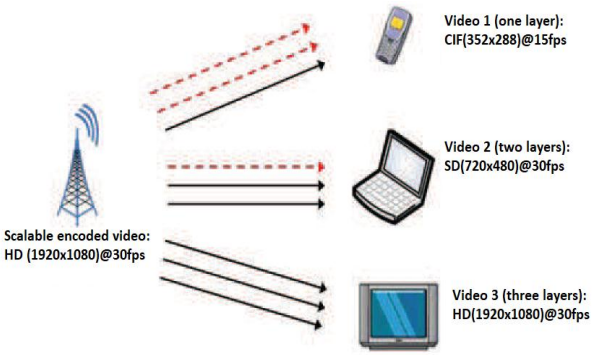



**Figure XXXI The Streaming Of The Encoded Layers And The Particular Layers Decoded At The Different Receivers [42]**

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

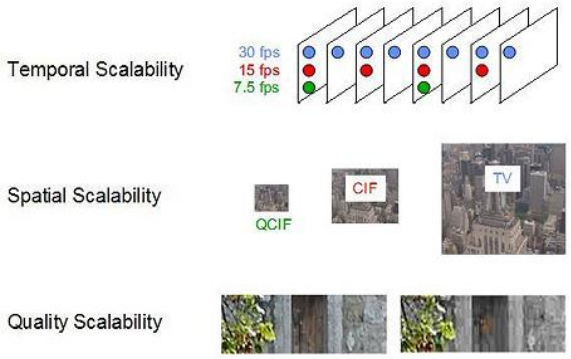



**Figure XXXII Basic types of scalable video coding [43]**

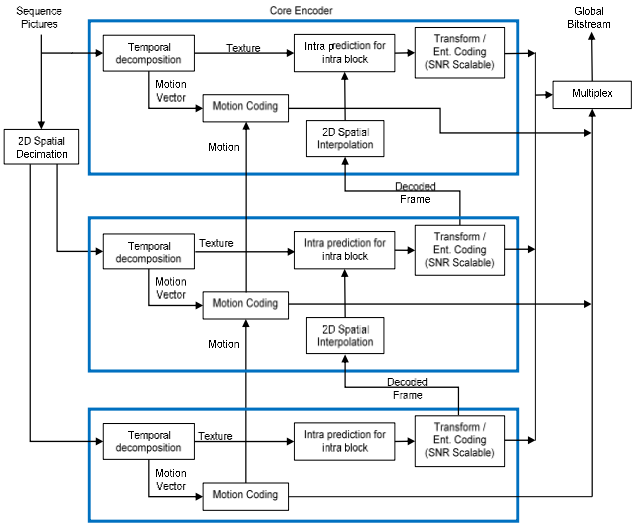



**Figure XXXIII Scalable video (with spatial, temporal and SNR scalabilities) coding structure with three coding layers [8]**

In terms of performance and complexity, dependent coding of layers is often compared against simulcast (independent coding of equivalent signals). Typical applications where scalable coding or simulcast would be applied, such as flexible rate or resolution switching, would usually only output one of the layers. However, in the case of multi-loop decoding, it is still necessary to decode all reference layers, such that the overall decoding complexity increases compared to simulcast. This effect is more critical in SNR scalability, where the reference layers are not subsampled. On the other hand, dependent coding of layers has advantages over simulcast in terms of compression performance.

Tables IV and V show the average bit rate savings for equal luma PSNR across four base layer QP values (22, 26, 30, and 34) [2]. Two types of comparisons are made, as shown in separate columns. A simulcast comparison is made, in which the enhancement layer (EL) plus base layer (BL) are compared to simulcast of a high-resolution single-layer bitstream at the same resolution as the enhancement layer plus the identical base layer.

Additionally, a comparison is made where only the enhancement layer is compared to the high-resolution single-layer bitstream. The latter number is deemed relevant for the cost savings when introducing an additional service based on scalable technology (e.g. Ultra-HD broadcast when an HD broadcast of the same program already exists). See Refs. [37]–[43]. The first frames from each test sequence in Tables XII and XIII are shown in Fig. XXXIV, and various picture's partitions are shown in Figs. XXXV - XXXVI.

**TABLE XII**
**BIT RATE REDUCTION OF SHVC VS. SIMULCAST: 2× SPATIAL SCALABILITY [2]**

| Sequence | ΔQP=0 | | ΔQP=2 | |
|---|---|---|---|---|
| | EL+BL vs. simulcast | EL only vs. high res. single layer | EL+BL vs. simulcast | EL only vs. high res. single layer |
| Kimono | 19.8% | 29.2% | 27.3% | 47.5% |
| ParkScene | 12.6% | 17.6% | 17.6% | 27.8% |
| Cactus | 11.6% | 16.6% | 16.7% | 27.7% |
| BasketballDrive | 14.5% | 19.9% | 20.8% | 33.0% |
| BQTerrace | 6.0% | 7.3% | 8.5% | 12.1% |
| Average | 12.9% | 18.1% | 18.2% | 29.6% |

**TABLE XIII**
**BIT RATE REDUCTION OF SHVC VS. SIMULCAST: 1.5× SPATIAL SCALABILITY [2]**

| Sequence | ΔQP=0 | | ΔQP=2 | |
|---|---|---|---|---|
| | EL+BL vs. simulcast | EL vs. high res | EL+BL vs. simulcast | EL vs. high res |
| Kimono | 29.0% | 49.5% | 40.7% | 78.1% |
| ParkScene | 22.3% | 36.0% | 31.7% | 58.8% |
| Cactus | 21.1% | 34.7% | 30.8% | 58.5% |
| BasketballDrive | 24.6% | 38.6% | 34.6% | 61.9% |
| BQTerrace | 13.5% | 18.0% | 20.3% | 33.3% |
| Average | 22.1% | 35.4% | 31.6% | 58.1% |

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

(a) A FRAME FROM TEST SEQUENCE Kimono1_1920x1080p_24 [28]


(b) A FRAME FROM SEQUENCE ParkScene_1920x1080p_24 [45] [28]

(c) A FRAME FROM SEQUENCE Cactus_1920x1080p_50 [45] [28]

(d) A FRAME FROM SEQUENCE BasketballDrive_1920x1080p_50 [45] [28]

(e) A FRAME FROM SEQUENCE BQTerrace_1920x1080p_60 [45] [28]

**FIGURE XXXIV THE FIRST FRAMES FROM EACH TEST SEQUENCE**

### *C. 3D Video Extensions*

3D and multiview video formats can enable depth perception for a visual scene when used with an appropriate 3D display system [2]. The available types of 3D displays include stereoscopic displays that are viewed with special glasses to enable the display of different views to each eye of the viewer, and auto-stereoscopic displays that emit view-dependent pixels and do not require glasses for viewing. The latter kind of displays often employ depth-based image rendering techniques, where it is desirable to use high-quality depth maps as part of the coded representation. Therefore, video plus depth is another important and emerging class of 3D formats.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

These can also allow for advanced stereoscopic processing, such as adjusting the level of depth perception in conventional stereo displays according to display size, viewing distance, user preference, etc. The depth information itself may be extracted from a stereo pair by solving for stereo correspondences or may be obtained directly through special range cameras; it may also be an inherent part of the content, e.g. in 3D computer graphics generated imagery. To support these applications, HEVC extensions for the efficient compression of stereo and multiview video were developed by JCT-3V, and the inclusion of depth maps to support advanced 3D functionalities was also included.

The most straightforward architecture is a multiview extension of HEVC that is referred to as MV-HEVC. It uses the same design principles of the prior MVC extension in the AVC framework [9], [10]. This extension of HEVC was finalized in July 2014, and the draft text can be found in [11] and [72].


**FIGURE XXXV A B-FRAME OF BasketballDrillText_832x480_50_qp22 [27]**

## VIII. VERSATILE VIDEO CODING

Versatile Video Coding (VVC) is a video coding standard by the Joint Video Experts Team (JVET) of ITU-T of SG 16 WP3 and the ISO/IEC JTC of 1/SC 29/WG organizations, for video compression currently under development which is due by June 2020[75]. The addition of new algorithms of VVC is projected to deliver 30% to 50% compression efficiency for the same video quality when compared to HEVC, with the inclusion of subjectively lossless and lossless compression. VVC supports resolutions from 4K to 16K. It also works with 360° video contents. VVC supports all the YUV formats, that is 4:4:4, 4:2:2 and 4:2:0 having 10 to 16 bits for each component.

Some of the other contents VVC supports are BT .2100 wide color gamut, the high dynamic range (HDR) with more than 16 stops (that is peak brightness of 1000, 4000 and 10000 nits). Other codec features it supports are auxiliary channels, frame rate with variable and fractional forms from 0 to 120 Hz, scalable video coding for spatial, temporal, dynamic range differences, color gamut, SNR, panoramic formats, stereo/Multiview coding and still picture coding. Encoding complexity is ten times more than that of HEVC. The decoding complexity is estimated to be at least twice that of HEVC [78].

VVC development is ongoing using the VVC Test Model (VTM). This VTM is a reference software codebase which has a minimal set of coding tools. Further coding tools are added after being tested in Core Experiments (CEs). Its predecessor was the Joint Exploration Model (JEM), an experimental software codebase that was based on the reference software used for HEVC. The latest version of the reference software (VTM-5.0) is available at https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM.
Table XIV shows the comparative study between HEVC and VVC of the algorithms used in different blocks in HM 16.18(HEVC) and BMS 1.0 or VTM (VVC) softwares. These algorithms are also shown in the VVC block diagram in Fig XXXVII.

**TABLE XIV**
**TOOL COMPARISON BETWEEN HEVC AND VVC [76]**

| | H.265/HEVC HM 16.18 | H.266/VVC BMS 1.0 (draft) |
|---|---|---|
| **Block Structure** | Quadtree<br>CTU size up to 64x64 | (QTBT) + Ternary Tree (TT)<br>CTU size up to 256x256 |
| **Intra Prediction** | 35 intra prediction modes. | 65 intra prediction modes with improved intra mode coding<br>Cross-component linear model (CCLM) prediction |
| **Inter prediction** | Hierarchical weighted prediction (P, B frames)<br>PU level motion vector prediction<br>Motion vector difference<br>1/4 pel MV accuracy<br>Block motion comp.<br>Translation motion prediction | Hierarchical weighted prediction (P, B frames)<br>Sub-CU based motion vector prediction<br>Adaptive motion vector precision<br>Affine motion prediction<br>Decoder-side motion vector refinement |
| **Transform** | Transform block size 8x8, 16x16, 32x32<br>DCT-II and DST-VII | Transform block sizes 4x4 up to 64x64<br>Adaptive multiple core transforms<br>Mode dependent non-separable secondary transforms (4x4) |
| **Loop filter** | Deblocking filter, SAO | Deblocking filter, SAO, Adaptive loop filter |
| **Entropy Coding** | CABAC | Modified CABAC (with Context modelling for transform coefficient levels) |

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)

## IX. CONCLUSIONS

In the family of Video coding standards, HEVC has the promise and potential to replace/supplement all the existing standards. While the complexity of the HEVC encoder is several times that of the H.264/AVC, the decoder complexity [15] is within the range of the latter. Researchers are still exploring about reducing the HEVC encoder complexity, mostly in the area of motion estimation and compensation. To enhance the standard, several extensions have been proposed and implemented. These extensions will further enhance the utility of the HEVC standard and broaden its range of applications. Much of the technology with respect to the extensions has been finalized and implemented as standard extensions in July 2014 and beyond. Due to applications such as Ultra High Definition (UHD) compression, High Dynamic Range (HDR) and 360 videos, the need for a newer codec was proposed in Oct 2015 and planned for its release in June 2020 as Versatile Video Coding (VVC). Versatile Video coding provides a wide range of application use along with the increase in the complexity of the Encoder and the Decoder. This provides an area of future research by reducing the complexity of the codec and maintaining the quality of the reconstructed video.

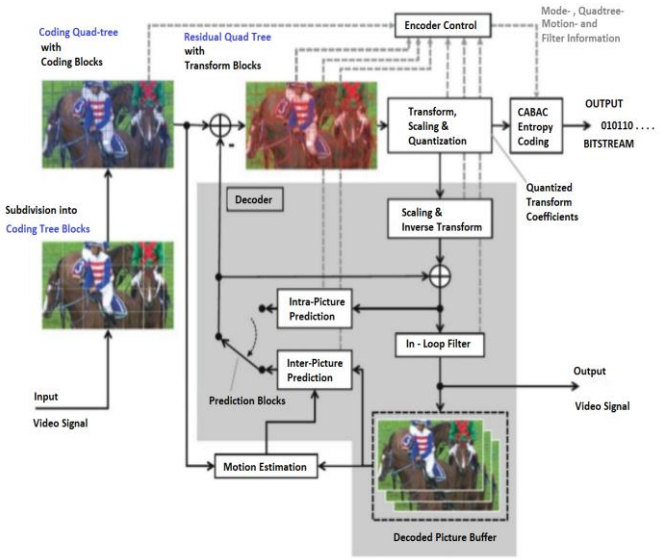



**Figure XXXVII Encoder Block Diagram of Versatile Video Coding**

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 9, Issue 12, December 2019)


**FIGURE XXXVI ANALYSIS OF A B-PICTURE IN FIGURE XXIII** (a) **PICTURE PARTITION DISPLAY AT CU LEVEL; STATIC AREAS ARE CODED WITH MAXIMUM CODING UNIT SIZE WHILE THOSE WITH HIGH MOTION ARE FURTHER DIVIDED INTO SMALLER BLOCKS TO IMPROVE PICTURE QUALITY,** (b) **PICTURE PARTITIONS INCLUDING CU, PU AND TU DISPLAY,** (c) **MOTION VECTORS OF THE PARTITIONED BLOCKS. THE BITSTREAM IS ANALYZED USING ELECARD HEVC ANALYZER © SOFTWARE [48].**

## REFERENCES


[1] G. J. Sullivan et al, "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.

[2] G.J. Sullivan et al, "Standardized Extensions of High Efficiency Video Coding (HEVC)", IEEE Journal of selected topics in Signal Processing, vol. 7, no. 6, pp. 1001-1016, Dec. 2013.

[3] D. Flynn, J. Sole, and T. Suzuki, "Range extensions draft 4" in Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVCN1005, 14th Meeting: Vienna, AT, Jul. 25–Aug. 2 2013.

[4] J. Chen, J. Boyce, Y. Ye, and M. M. Hannuksela, "Scalable high efficiency video coding draft 3," in Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-N1008, 14th Meeting: Vienna, Austria, July 25–Aug. 2 2013.

[5] S. Lui et al, "Video Prediction Block Structure and the Emerging High Efficiency Video Coding Standard", IEEE proceedings on Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, pp. 1-4, 2012.

[6] T. Wiegand et al, "Overview of the H.264/AVC video coding standard", IEEE Trans. on Circuits and Systems for Video Technology, vol.13, no.7, pp. 560-576, March 2003.

[7] C. Fogg, "Suggested figures for the HEVC specification", ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC- J0292r1, July 2012.

[8] Z. Wang, J. Zhang and H. Li, "Spatially scalable video coding with an efficient two-layered architecture", (online book) Springer 2009.

[9] Y. Chen et al, "The emerging MVC standard for 3D video services" EURASIP J. Adv. Signal Process., vol. 2009, no. 1, Jan. 2009, Article 8.

[10] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/AVC standard", Proc. IEEE, vol. 99, no. 4, pp. 626–642, Apr. 2011.

[11] G. Tech, K. Wegner, Y. Chen, M. M. Hannuksela, and J. Boyce, "MVHEVC draft text 5," in Joint Collaborative Team on 3D Video Coding Extensions (JCT-3V) Document JCT3V-E1004, 5th Meeting: Vienna, Austria, Jul. 27–Aug. 2, 2013.

[12] HEVC white paper: http://www.ateme.com/an-introduction-to-uhdtv-and-hevc

[13] HEVC tutorial by I.E.G. Richardson: http://www.vcodex.com/h265.html

[14] K. McCann et al, "HM9: High Efficiency Video Coding (HEVC) Test Model 9 Encoder Description", ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-K1002v2, Oct. 2012.

[15] F. Pescador et al, "Complexity analysis of an HEVC decoder based on a digital signal processor", IEEE Trans. on Consumer Electronics, vol. 59, pp. 391-399, May 2013.

[16] B. Bross et al, "High Efficiency Video Coding (HEVC) Text Specification Draft 10", ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), document JCTVC-L1003, Mar. 2013

[17] N. Ahmed, T. Natarajan, K.R. Rao, "Discrete Cosine Transform", IEEE Transactions on Computers, Vol. C-23, pp. 90-93, Jan. 1974.

[18] I.E.G. Richardson, "The H.264 advanced video compression standard", 2nd Edition, Hoboken, NJ, Wiley, 2010.

[19] I.E.G. Richardson, "Video Codec Design: Developing Image and Video Compression Systems", Wiley, 2002.

[20] K.R. Rao, D.N. Kim and J.J. Hwang, "Video Coding Standards: AVS China, H.264/MPEG-4 Part 10, HEVC, VP6, DIRAC and VC-1", Springer, 2014.

[21] I.K. Kim et al, "Block partitioning structure in the HEVC standard," IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, pp. 1697-1706, Dec. 2012.

[22] S. Liu and S. Lei, "Video Prediction Block Structure and the Emerging High Efficiency Video Coding Standard", IEEE Proc. on Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, pp. 1-4, 2012.

[23] M.T. Pourazad et al, "HEVC: The new gold standard for video compression," IEEE CE Magazine, pp. 36-46, vol. 1, issue 3, July 2012.

[24] F. Bossen et al, "HEVC Complexity and Implementation Analysis", IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1685-1696, Dec. 2012.

[25] K. Iguchi et al, "HEVC Encoder for Super Hi-Vision", 2014 IEEE International conference on Consumer Electronics (ICCE), pp. 61-62, 2014.

[26] J.-R. Ohm et al, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)", IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1669-1684, Dec. 2012.

[27] HEVC encoded bit streams: ftp://ftp.kw.bbc.co.uk/hevc/hm-11.0-anchors/bitstreams/

[28] Video Sequence Download Link- http://media.xiph.org/video/derf/

[29] HEVC Fraunhofer site containing all the information on HEVC- http://hevc.info/

[30] HM Software Manual – https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-9.2-dev/doc/software-manual.pdf

[31] HM 14.0 (HEVC Software) Download Link- https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/HM-14.0-dev/, (HM16.0)

[32] HEVC decoder for handheld devices implemented by Ace Thought – http://www.acethought.com/index.php/products/hevc-decoder/

[33] Special Issue on emerging research and standards in next generation video coding, IEEE Trans. on Circuits and Systems for Video Technology, vol. 23, no. 12, pp. 2009-2142, Dec. 2013.

[34] K.R. Rao and J.J. Hwang, "Techniques and standards for image/video/audio coding," Prentice-Hall, 1996.

[35] Moto, "HEVC - What are CTU, CU, CTB, CB, PB and TB?" CODE: Sequoia, worgpress.com site, blog. [online]. Available: http://codesequoia.wordpress.com/2012/10/28/hevc-ctu-cu-ctb-cb-pb-and-tb/ (accessed on June 9th 2014).

[36] S. Riabstev, "Detailed overview of HEVC/H.265", [online]. Available: https://app.box.com/s/rxxxzr1a1lnh7709yvih (accessed on June 12 2014).

[37] T. Hinz et al, "An HEVC Extension for Spatial and Quality Scalable Video Coding", Proceedings of SPIE, vol. 8666, pp. 866605-1 to 866605-16, Feb. 2013.

[38] SHVC software and software manual: The source code for the software and its manual is available in the following SVN repository. [online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/ (accessed on June 26th 2014).

[39] Test sequences for scalable video coding. [online]. Available: ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/ (accessed on June 26, 2014).

[40] J. Boyce et al, "Information for HEVC scalability extension", JCT-VC, Document JCTVC-G078, Nov. 2011. [online] Available: http://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=3327 (accessed on July 2, 2014).

[41] P. Heke et al, "A scalable video coding extension of HEVC", IEEE DCC, pp. 201-210, Mar. 2013.

[42] I. Unanue et al, "A Tutorial on H.264/SVC Scalable Video Coding and its Tradeoff between Quality, Coding Efficiency and Performance", Recent Advances on Video Coding. [online] Available: http://www.doc88.com/p-516795349043.html (accessed on June 20 2014).

[43] H. Schwarz and T. Wiegand, "The Scalable Video Coding Amendment of the H.264/AVC Standard", csdn.net, world pharos, blog. [online] Available: http://blog.csdn.net/worldpharos/article/details/3369933 (accessed on June 20th 2014).

[44] K. Choi and E.S. Jang, "Fast coding unit decision method based on coding tree pruning for high efficiency video coding," SPIE OE, vol. 51, issue 3, pp. 030502-1 – 030502-3, Mar. 2012.

[45] F.D. Simone et al, "Towards high efficiency video coding: Subjective evaluation of potential coding technologies," J. Visual Communication Image Representation, vol. 22, pp. 734–748, Nov. 2011.

[46] O.C. Au, HEVC: http://www.apsipa2013.org/wp-content/uploads/2013/09/Tutorial_8_NextGenerationVideoCoding_Part_2.pdf

[47] S. Vasudevan and K.R. Rao, "Combination method of fast HEVC encoding," IEEE ECTICON 2014, Korat, Thailand, May 2014.

[48] Elecard Software can be found at: http://www.elecard.com/

[49] M. Naccari et al, "Improving Inter Prediction in HEVC with Residual DPCM for Lossless Screen Content Coding" IEEE Picture Coding Symposium, pp. 361-364, Dec. 2013.

[50] M. Budagavi and D.K. Kwon, "Intra motion compensation and entropy coding improvements for HEVC screen content coding", IEEE Picture Coding Symposium, pp. 365-368, Dec. 2013.

[51] JCT-VC Document management system – Sapporo, Valencia meetings. http://phenix.int-evry.fr/jct/index.php

[52] Lossless video coding: http://web.stanford.edu/class/ee398b/handouts/lectures/LosslessVideoCoding.pdf

[53] M. Dai, D. Loguinov and H.M. Radha, "Rate-Distortion Analysis and Quality Control in Scalable Internet Streaming", IEEE Trans. on Multimedia, vol. 8, no. 6, pp. 1135–1146, Dec. 2006.

[54] M. Zhou et al, "HEVC lossless coding and improvements", SPIE/EI, vol. 8666-10, Burlingame, CA, Feb. 2013.

[55] M. Zhou et al, "HEVC lossless coding and improvements," IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1839–1843, Dec. 2012.

[56] A. Norkin and G. Bjontegaard, "HEVC Deblocking Filter," IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1746–1754, Dec. 2012.

[57] F. Chih-Ming and E. Alshina, "Sample adaptive offset in the HEVC Standard," IEEE Trans. On Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1755–1764, Dec. 2012.

[58] Y.L. Lee, K.-H. Han and G. J. Sullivan, "Improved lossless intra coding for H.264/MPEG-4 AVC," IEEE Trans. on Image Process., vol. 15, no. 9, pp. 2610–2615, Sep. 2006.

[59] M. Rerabek and T. Ebrahimi, "Comparison of compression efficiency between HEVC/H.265 and VP9 based on subjective assessments," SPIE Optical Engineering + Applications, San Diego, CA, Aug. 18-21, 2014. (The conference proceedings include few papers on HEVC. Also VP9 and H.264)

[60] D. Grois et al, "Comparative assessment of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders for low-delay video applications," SPIE Optical Engineering + Applications, San Diego, CA, Aug. 18-21, 2014.

[61] A. Heindel, E. Wige and A. Kaup, "Sample-based weighted prediction for lossless enhancement layer coding in HEVC," IEEE PCS 2013, Grand Compression Challenge, 2013.

[62] A. Heindel, E. Wige and A. Kaup, "Analysis of prediction algorithms for residual compression in a lossy to lossless scalable video coding system based on HEVC," SPIE Optical Engineering, Applications of digital image processing, vol. 9217, San Diego, California, USA, Aug. 18-21, 2014.

[63] A. Lee et al, "Efficient inter prediction mode decision method for fast motion estimation in High Efficiency Video Coding," ETRI Journal, vol. 36, pp. 528-536, Aug. 2014.

[64] X. Li and M.T. Orchard, "Edge-directed prediction for lossless compression of natural images," IEEE Trans. on Image Process. vol. 10, pp. 813–817, June 2001.

[65] Z. Lv and R. Wang, "An all zero blocks early detection method for high efficiency video coding" SPIE, vol. 9029, pp. 902902-1 thru 902902-7, IS&T/SPIE Electronic Imaging, San Francisco, CA, Feb. 2014.See [66, 67].

[66] P-T. Chiang and T.S. Chang, "Fast zero block detection and early CU termination for HEVC video coding", IEEE ISCAS 2013, pp.1640-1643, Beijing, China, May 2013. See [65, 67].

[67] K. Lee et al, "A novel algorithm for zero block detection in high efficiency video coding", IEEE Journal of selected topics in signal processing, vol. 7, #6, pp. 1124-1134, Dec. 2013. See [65, 66].

[68] V. Sze and M. Budagavi, HEVC tutorial, website on ISCAS 2014, http://iscas2014.org/

[69] HEVC Fraunhofer site containing all the information on HEVC-http://hevc.info/ (software/overview papers/documents etc.)

[70] M.J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I A low complexity context-based, lossless image compression algorithm," IEEE Proc. of Data Compression Conference, pp. 140-149, Snowbird, Utah, Mar. 1996.

[71] I.G. Kim et al., High Efficiency Video Coding (HEVC) Test Model 15 (HM15) Encoder Description, JCTVC-Q1002, April 2014.

[72] J. Boyce et al. "Draft high efficiency video coding (HEVC) version 2, combined format range extensions (RExt), scalability (SHVC), and multi-view (MV-HEVC) extensions", JCT-VC, July 11, 2014.

[73] Ochoa Domínguez, H. and Rao, K.R., 2018. Versatile Video Coding. River Publishers

[74] Ochoa-Dominguez, H. and Rao, K.R., 2019. Discrete Cosine Transform. II Edition, CRC Press.

[75] B. Bross, et al, "Versatile Video Coding (Draft 2)", ISO/IEC doc. JVET-K1001, Ljubljana, SL, 10–18 July.

[76] P. Topiwala, M. Krishnan, W. Dei, "Performance Comparison of VVC, AV1 and HEVC on 8-bit and 10-bit content", SPIE Int'l Symposium, San Diego, CA, Aug., 2018.

[77] Topiwala, Pankaj, Madhu Krishnan, and Wei Dai. "Performance comparison of VVC, AV1 and EVC." In Applications of Digital Image Processing XLII, vol. 11137, p. 1113715. International Society for Optics and Photonics, 2019.

[78] Versatile video coding wiki page. Available: https://en.wikipedia.org/wiki/Versatile_Video_Coding

[79] VVC Fraunhofer site containing all the information on VVC - https://jvet.hhi.fraunhofer.de/

[80] VTM Version 1.1 (VVC Software) subversion page: https://jvet.hhi.fraunhofer.de/svn/svn_VVCSoftware_VTM/

[81] JVET document site for VVC: http://phenix.it-sudparis.eu/jvet/

[82] Jens-Rainer, Ohm and Gary J. Sullivan. "Versatile Video Coding–towards the next generation of video compression." In Picture Coding Symposium 2018. 2018.

[83] Pan, Zhaoqing, He Qin, Xiaokai Yi, Yuhui Zheng, and Asifullah Khan. "Low complexity versatile video coding for traffic surveillance system." International Journal of Sensor Networks 30, no. 2 (2019): 116-125.

[84] Wieckowski, Adam, Jackie Ma, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. "Fast partitioning decision strategies for the upcoming versatile video coding (VVC) standard." In 2019 IEEE International Conference on Image Processing (ICIP), pp. 4130-4134. IEEE, 2019.

[85] Abdoli, Mohsen, Felix Henry, Patrice Brault, Pierre Duhamel, and Frédéric Dufaux. "Short-distance intra prediction of screen content in versatile video coding (VVC)." IEEE Signal Processing Letters 25, no. 11 (2018): 1690-1694.

[86] Kammoun, Ahmed, Wassim Hamidouche, Fatma Belghith, Jean-François Nezan, and Nouri Masmoudi. "Hardware design and implementation of adaptive multiple transforms for the versatile video coding standard." IEEE Transactions on Consumer Electronics 64, no. 4 (2018): 424-432.

[87] Azgin, Hasan, Ercan Kalali, and Ilker Hamzaoglu. "An Efficient FPGA Implementation of Versatile Video Coding Intra Prediction." In 2019 22nd Euromicro Conference on Digital System Design (DSD), pp. 194-199. IEEE, 2019.

[88] Kammoun, Ahmed, Wassim Hamidouche, Pierrick Philippe, Fatma Belghith, Nouri Massmoudi, and Jean-François Nezan. "Hardware Acceleration of Approximate Transform Module for the Versatile Video Coding Standard." European Signal Processing Conference (EUSIPCO 2019).

[89] Garrido, Matías J., Fernando Pescador, M. Chavarrías, P. J. Lobo, and Cesar Sanz. "A 2-D multiple transform processor for the versatile video coding standard." IEEE Transactions on Consumer Electronics 65, no. 3 (2019): 274-283.

[90] VVC tutorial on slide share: https://www.slideshare.net/mohsenirib/an-introduction-to-versatile-video-coding-vvc-for-uhd-hdr-and-360-video-135899487