5-24-2018

Non-MPM Mode Coding for Intra Prediction in Video Coding

Taru Kanchan
Santa Clara University, tkanchan@scu.edu

Santa Clara University
Department of Computer Engineering

Date: 5/24/2018

I recommend that the following thesis prepared under my supervision by

Taru Kanchan

titled

Non-MPM Mode Coding for Intra Prediction in Video Coding

to be accepted in partial fulfillment for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Thesis Advisor Dr. Nam Ling

Thesis Reader Dr. Minqiang Jiang

Chair of Department Dr. Nam Ling

Intra Mode Coding for Video Coding

by

Taru Kanchan

COEN 497

Master's Thesis Research

Thesis

Submitted in partial fulfillment of requirements

for the degree of Master of Science

in Computer Science and Engineering

in the School of Engineering at

Santa Clara University, 2016

Santa Clara, California, USA

Date: 5/24/2018

To my family

Abstract

The High Efficiency Video Coding standard introduced thirty-five intra prediction modes. It employed a method based on three most probable modes (MPM) to improve intra mode coding. This method significantly improved the performance by extracting three MPMs out of the thirty-five intra modes. The Joint Video Exploration Team (JVET) defines sixty-seven intra prediction modes for a possible future video coding standard. In the latest JVET development, six MPMs are chosen, and the remaining sixty-one modes are divided into sixteen “selected” and forty-five “non-selected” modes. These non-MPM modes are coded using fixed length coding. This research focusses on finding more efficient ways to code these intra prediction modes, including MPM modes and non-MPM modes. A method is proposed to select and order the sixty-one non-MPM modes based on probability statistics. The modes that fall into selected category are coded using shorter codes and non-selected modes are coded using larger codes, which is in line with the principle of entropy coding. Experimental results prove performance improvement when compared to JEM7.0 software as a reference.

Keywords *- video coding, visual communications, HEVC, JEM, most probable modes, intra prediction, non-MPM, intra mode coding.*

Acknowledgements

Foremost, I would like to express my sincere gratitude to Dr. Nam Ling and Dr. Minqiang Jiang for their continuous support and guidance throughout the duration of my research. Their constant motivation and encouragement kept me going, when I felt nothing was working out. I could not have imagined a better advisor and mentor for my research.

My sincere thank goes to the Santa Clara University for providing me all day access to the Video research lab which helped me work flexibly on my own schedule.

Finally I would like to thank my amazing family and friends for encouraging me and supporting me in this journey. This would not have been possible without you all.

Table of Contents

List of Figures

List of Tables

1

1. Introduction

1.1 Need for improved coding techniques

With the widespread use of smartphones and improving bandwidth and storage mediums, video applications are becoming more and more popular. According to the report published by Cisco [1] in June 2017 in their annual Visual Network Index (VNI) Forecast, by 2021 the total global IP (Internet Protocol) traffic will account for 82% of all internet traffic. Out of this around 13% will be live video traffic. Video related applications such as video surveillance, Video on Demand (VoD), and virtual reality (VR) applications are bound to increase manifold. The Content Delivery Network (CDN) will be carrying 71% of all internet traffic. With such growing demand for video and increasing video resolution and quality, it is imperative to design efficient applications, especially since the bandwidth is not growing at the same rate as video content. Video compression techniques, as such, need continuous refinement to account for the changing scenario. Some of the statistics published by Cisco demonstrating current and predictions for future trends are shown in Figure 1.1, Figure 1.2 and Figure 1.3.

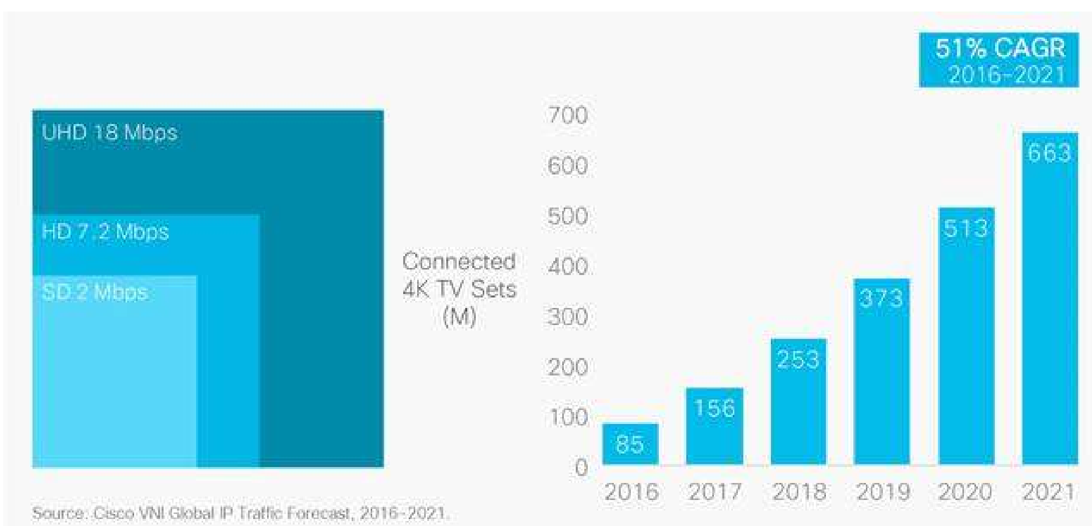


Figure 1.1: Trend for increasing 4K TV sets [1]

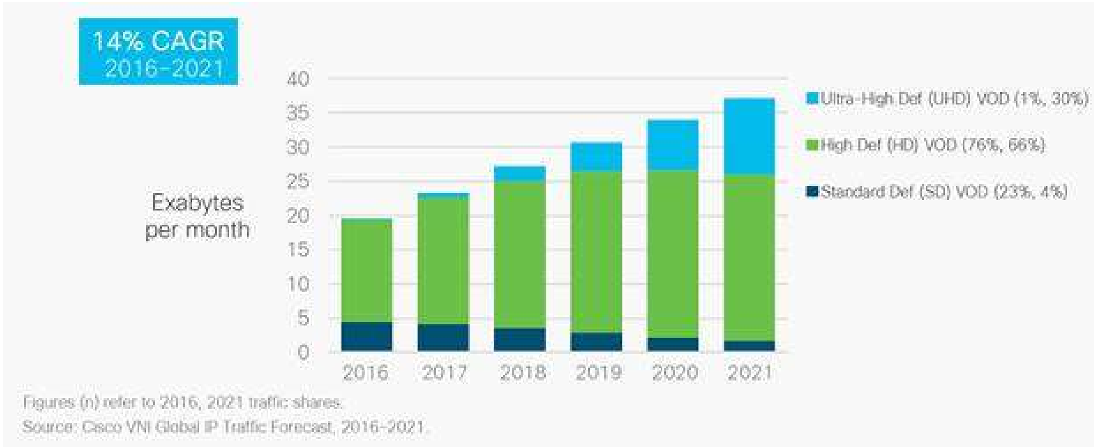


Figure 1.2: Global 4K VoD traffic [1]

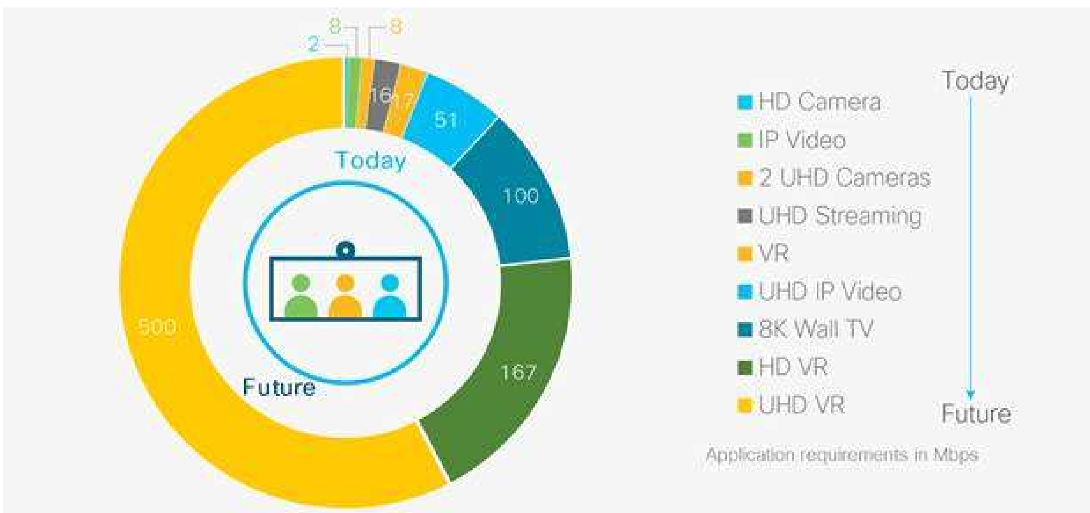


Figure 1.3: Demand for video in homes [1]

1.2 History of video coding standards

The H.264/Advanced Video Coding (AVC) standard, which was first released in 2003, resulted in significant compression and quality improvements as compared to the previous standards [13]. With the advent of H.264/AVC standard, there have been many significant changes in the industry. The digital High Definition Television (HDTV) became popular and has now replaced the analog Standard Definition Television (SDTV) [2]. Streaming internet videos and video conferencing have become mainstream. The H.264/AVC standard employed many new techniques and provided significant bit rate improvement from previous standards. In other words, if the same amount of bandwidth is used, this standard provided much better picture quality as compared to its predecessors.

To keep up with the increasing popularity of high resolution video content, the Joint Collaborative Team on Video Coding (JCT-VC) decided to work on a new video coding standard. The JCT-VC was a partnership between the ITU-T (International Telecommunication Union – Telecommunication Standardization Sector) Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG), and they came up with the High Efficiency Video Coding (HEVC) standard in 2013. The HEVC standard showed 50% bit rate improvement as compared to its predecessor H.264/AVC [14]. There were many significant changes in the HEVC standard which include partitioning, intra prediction and inter prediction. The JCT-VC was even awarded the Primetime Emmy Engineering Award for outstanding achievement of developing this standard [3][4]. The HEVC standard was recognized as the primary coding format for Ultra High Definition (UHD) TV.

In October 2015 the ITU-T VCEG and ISO/IEC MPEG agreed to collaborate together and formed the Joint Video Exploration Team (JVET) with a goal of working for a possible future video coding standard. The Joint Exploration Model (JEM) is the reference software for the JVET group [5]. The work for this possible future video coding standard is in progress. This research is based on this possible standard and the work uses JEM software as a base.

1.3 Research area and thesis outline

One of the major areas which have undergone considerable amount of change over the years in video coding is Intra Mode Coding. Coding of these intra prediction modes is based on the principle of entropy coding. According to this principle, the information which has a high probability of occurrence is coded using shorter codes, while the information which has lower probability of occurrence is coded using longer codes. This ensures that the bit rate is low and coding efficiency is high.

The H.264/AVC supported nine intra prediction modes [8]. Out of these nine modes, eight were angular prediction modes and one was DC mode. Angular prediction uses pixels directly above and left to the current block. H.264/AVC employed one most probable mode (MPM) based method to code these intra prediction modes. In this method, one mode which has the highest probability of occurrence is selected to be the most probable mode [8] . This method improved coding efficiency.

With increasing resolution and larger block sizes, eight directional modes were not sufficient. To improve directional prediction accuracy, HEVC came up with thirty-five intra prediction modes [6]. Thirty-three of these are angular prediction modes and the rest two are DC mode and Planar mode. The DC mode and the Planar mode uses average values to generate smooth samples [6]. The DC mode uses average of all samples, whereas the Planar mode is a little different and uses average of two linear predictions [6]. The one MPM method of H.264/AVC was modified to three MPM method in HEVC, where three most probable modes are selected and are coded using one or two bits. The remaining thirty-two modes are coded using a fixed length code [2].

The JEM software supports sixty-seven intra prediction modes. This includes DC mode, Planar mode and sixty-five angular prediction modes [15]. In the first version of this software, a simple extension of HEVC is used. The three MPM method of HEVC has been modified to include six MPMs. This leaves remaining sixty-one modes.

This research revolves around coding of these intra prediction modes. There have been many proposed methods to code these modes efficiently. This project proposes a new method based on probability statistics to code intra prediction modes to improve coding efficiently.

The structure of this document is as follows.

- The next section discusses the literature review which was done for the purpose of this research. A few relevant existing and proposed methods are discussed.
- The third section describes in detail the experiments performed, the theory behind that followed by the experimental results. A new method for intra mode coding is proposed which is discussed at length.
- The fourth section concludes this report.

2

2. Literature

This chapter discusses the design of HEVC standard which is required for understanding of this research. This is followed by intra prediction techniques and intra coding methods that were used in H.264/AVC standard, followed by the modifications and new methods adopted in HEVC. It then provides an overview of the various methods that have been proposed for improving intra coding for a possible future video coding standard.

2.1 Understanding HEVC

HEVC is the successor to the highly popular and widely used H.264 video coding standard, also known as AVC (Advanced Video Coding). With the emergence of high resolution video content, especially beyond-HD formats, there is a need for higher compression ratio (or less bit rate) to maintain same video quality as H.264. HEVC supports beyond-HD format resolutions (4kx2k and 8k*4k resolution) and offers double the compression efficiency than H.264 while providing same video quality level [14]. If the bit rate is kept same as H.264, it offers considerable improvement in video quality.

The focus while developing HEVC standard was not only improved coding efficiency, but also support for parallel processing. There are a number of features to support this. This includes wavefront parallel processing, support of partitioning a picture in tiles and dependent slice segments [6].

2.1.1 Network Abstraction Layer

A typical HEVC bitstream consist of logical packets called NAL (network abstraction layer) units. A bitstream start with VPS (video parameter set), SPS (sequence parameter set), PPS (picture parameter set) followed by slices of video data. The first slice is an IDR slice, followed by I, P or B slices [6].

The basic syntax element of the HEVC bitstream, similar to H.264, is the NAL unit. It is a logical packet with a header and a payload. The header is a two byte information, conveying its type and purpose in the bitstream. All NAL units fall into two broad categories differentiating the kind of information being carried by the NAL unit.

i. VCL (video coding layer) NAL unit
ii. Non-VCL NAL unit

The VCL NAL unit contain coded video data from a slice segment, whereas the non-VCL NAL units contain metadata and control information which is required by the decoder to decode the bitstream. The information in non-VCL NAL units can be associated with multiple coded pictures. Parameter sets are an example of non-VCL NAL unit [16]. The different kinds of NAL units in HEVC are provided in the table below.

Type	Meaning	Class
0, 1	Slice segment of ordinary trailing picture	VCL
2, 3	Slice segment of TSA picture	VCL
4, 5	Slice segment of STSA picture	VCL
6, 7	Slice segment of RADL picture	VCL
8, 9	Slice segment of RASL picture	VCL
10–15	Reserved for future use	VCL
16–18	Slice segment of BLA picture	VCL
19, 20	Slice segment of IDR picture	VCL
21	Slice segment of CRA picture	VCL
22–31	Reserved for future use	VCL
32	Video parameter set (VPS)	non-VCL
33	Sequence parameter set (SPS)	non-VCL
34	Picture parameter set (PPS)	non-VCL
35	Access unit delimiter	non-VCL
36	End of sequence	non-VCL
37	End of bitstream	non-VCL
38	Filler data	non-VCL
39, 40	SEI messages	non-VCL
41–47	Reserved for future use	non-VCL
48–63	Unspecified (available for system use)	non-VCL

Table 2.1: NAL unit types [6]

2.1.2 Parameter Sets

Before H.264 losing a packet containing sequence header or picture header would mean losing the entire information for that picture or that GOP (Group Of Pictures). To combat this serious issue, parameter sets were introduced in H.264. HEVC contains an additional parameter set called VPS [16]. The parameter sets consist of information which the decoder uses for decoding process. Parameter sets have the flexibility that they need not be a part of the bitstream. They can be transmitted to the decoder separately. The decoder references to the required parameter set using an identification. The three types of parameter sets are

i. VPS – Video Parameter Set
ii. SPS – Sequence Parameter Set
iii. PPS – Picture Parameter Set

The picture parameter set carries information required to decode a picture and hence the slices within a certain picture refer to the same PPS. Similarly the SPS carries information which is common to all the slices in the current sequence. The VPS carries information which is common to all video coding layers [16].

2.1.3 Supplemental Enhancement Information (SEI) and Video Usability Information (VUI)

These units are used for carrying information which is not needed by the decoder for the decoding process but for some other supplemental uses, such as timing of the decoding pictures, detecting losses and frame packing information [16].

The Figure 2.1 below shows the typical structure of a HEVC bitstream, starting from VPS, followed by SPS, PPS and slices of video coded data.

VPS	SPS	PPS	Slice	Slice	Slice

Figure 2.1: The HEVC bitstream structure

2.1.4 Video Coding Layer

The video coding layer of HEVC is very similar to H.264, with the exception of a few new introduced coding tools to improve coding efficiency. One such tool is the introduction of quad-tree based coding unit structure. This is discussed in detail in the next section.

The first slice of a sequence and the slice at random access points need to be coded in intra mode since they cannot refer to another slice. In intra mode coding, spatial correlation is utilized within a picture with no dependence on any previously decoded picture [6].

Pictures which are not required to be decoded independently utilize temporal correlation among neighboring pictures and are coded with interpicture prediction. Each block is motion compensated and its motion vectors (MV) along with coded residual data and mode decision data are sent to the decoder. The coded residual data contain the difference between the original data and the prediction made. This information is then transformed and coded using entropy coding [6]. The Figure 2.2 below shows the block diagram of HEVC encoder.

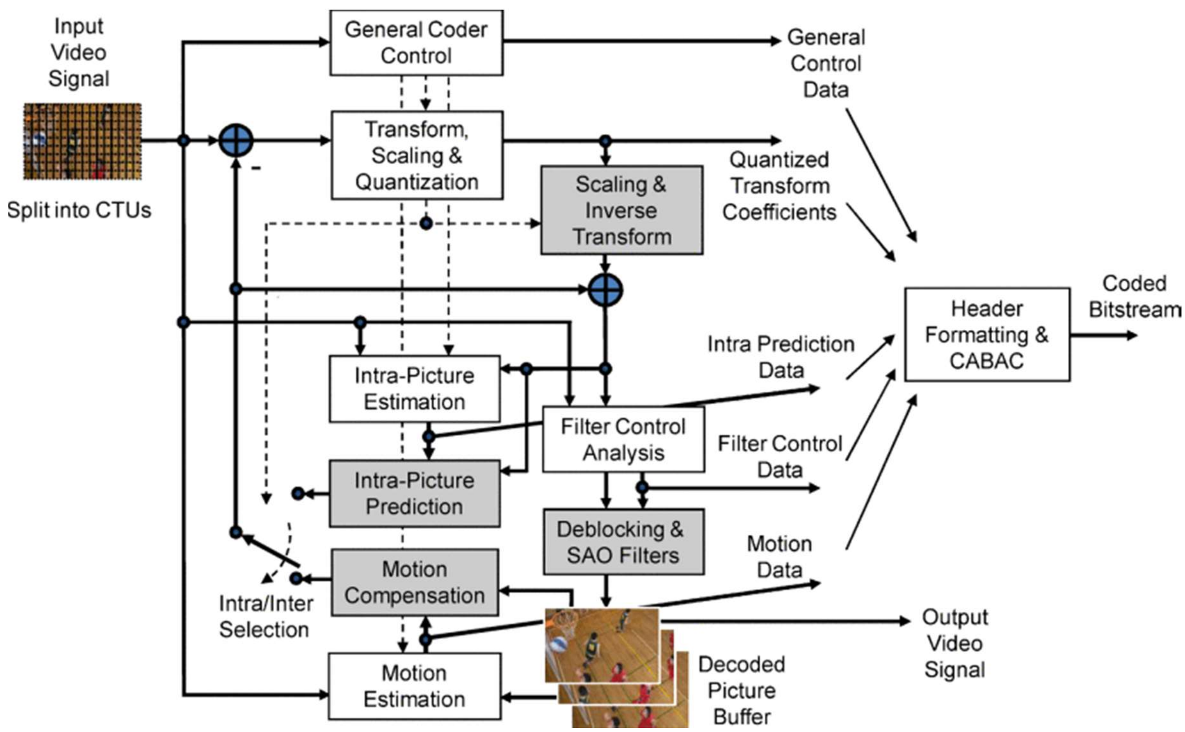


Figure 2.2: HEVC encoder block diagram [6]

2.1.5 Picture partitioning

Coding Tree Units and Coding Tree Blocks

Although the basic architecture is similar to H.264, HEVC has introduced some new coding tools, one of which is the coding unit quadtree structure [6]. This structure provides the flexibility of dividing a picture into smaller units of variable sizes. These basic units, into which the picture is divided are called coding tree units (CTU). The size of CTUs in a sequence is fixed and is passed as a parameter in the sequence parameter set. The CTU size can be 16x16, 32x32 or 64x64. Each CTU consist of a luma CTB (coding tree block) and a chroma CTB, along with syntax information, where the size of CTB is same as the size of CTU [1].

A slice is, thus, a sequence of CTUs in raster scan order. Each slice is independent in terms of intra prediction. Prediction is not allowed across slice boundaries [6].

Coding Units and Coding Blocks

Each CTB may contain a single CU (coding unit) or multiple CUs, and each CU consist of a luma coding block (CB), a chroma CB and syntax elements required by decoder. The size of a CU can be 64x64, 32x32,16x16 or 8x8 blocks [6]. A CU is the point where the decision of the type of prediction is made and transmitted. Depending on whether the prediction type is intra mode or inter mode, it may further be partitioned into prediction blocks (PB) and transform blocks (TB) [6].

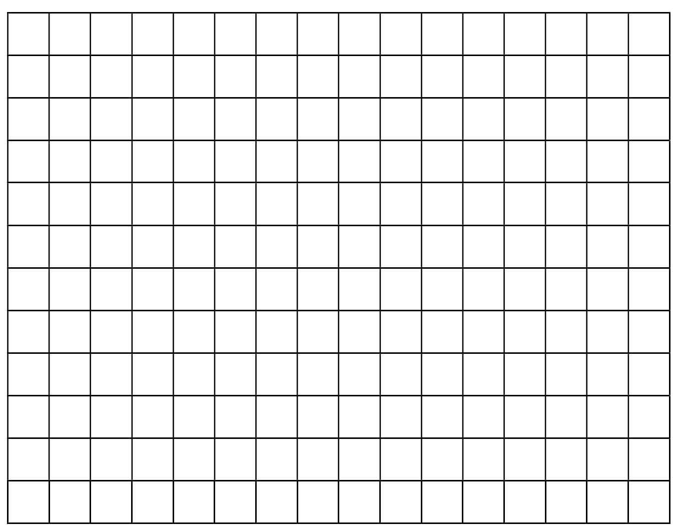


Figure 2.3: A picture partitioned into CTUs

2.1.6 Prediction Blocks and Transform Blocks

A PB is a partitioned unit within a CB in which same prediction is applied. Since a CB can be too big to be predicted as a whole block, it is partitioned depending on the content and type of predictability information. There are eight different partition modes for a PB [6].

Figure 2.4 shows these eight different partition modes for a PB.

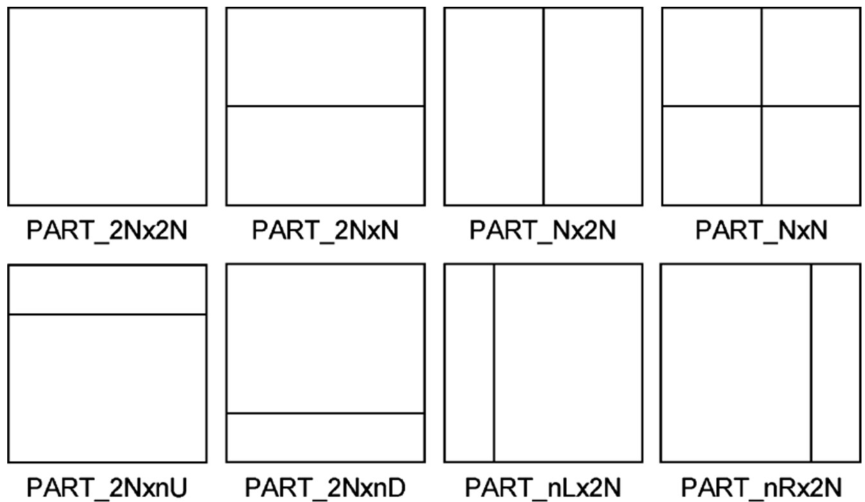


Figure 2.4: Eight PB partition modes [17]

Similarly a CB may be divided into transform blocks (TB) for coding the residual information using transforms. Transformation and quantization is applied at this level. This TB partition also depends on the content and may be different than the PB partition. Since transforms used in HEVC are square based, TBs can only be squares ranging from 4x4 upto 32x32. In case of intra mode prediction, a TB cannot be larger than a PB, whereas in case of inter picture prediction, this is possible and often makes the transform and quantization process more practical [6].

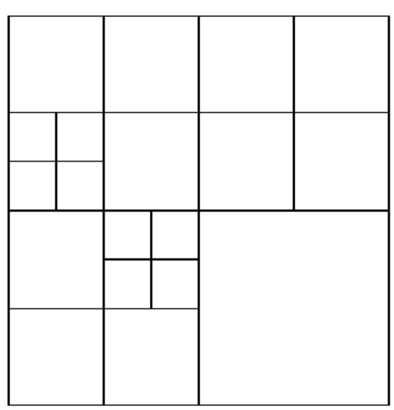


Figure 2.5: Division of CTU in CU and TU

2.1.7 Interpicture Prediction

HEVC introduced more adaptive motion parameters for motion compensation which are much more accurate than H.264 [6]. It allows sub pixel prediction.

Inter picture prediction mode utilizes temporal correlation between consecutive pictures. Inter prediction has been vastly improved in HEVC with the introduction of merge mode. The encoder has a choice to use merge more or directly code motion parameters.

In the merge mode, the inter-coded neighbors of the current PU are selected. The best motion parameters from these candidates are then used to infer the current PU. These candidates can be both from spatial neighbors or temporal neighbors. For a particular neighbor, only its index is coded, along with the difference in motion parameters [6].

2.1.8 Parallel Processing

HEVC provides support for parallel processing in two ways:

i. **Tiles**
HEVC provides capability for a picture to be divided into rectangular tiles. These tiles are approximately same in size each of which can be decoded independently. They can thus be processed in parallel [6]. Because of the use of rectangular tiles, this method can introduce visual artifacts at boundaries.

ii. **Wavefront parallel processing**
Wavefront parallel processing allows a slice to be partitioned as rows of CTUs and each row can be handled independently by a separate thread. The condition is that the thread processing row “n+1” can only start when row “n” has processed two CTUs. This method avoid visual artifacts of the first method and offers parallelism at a finer level of granularity [6].

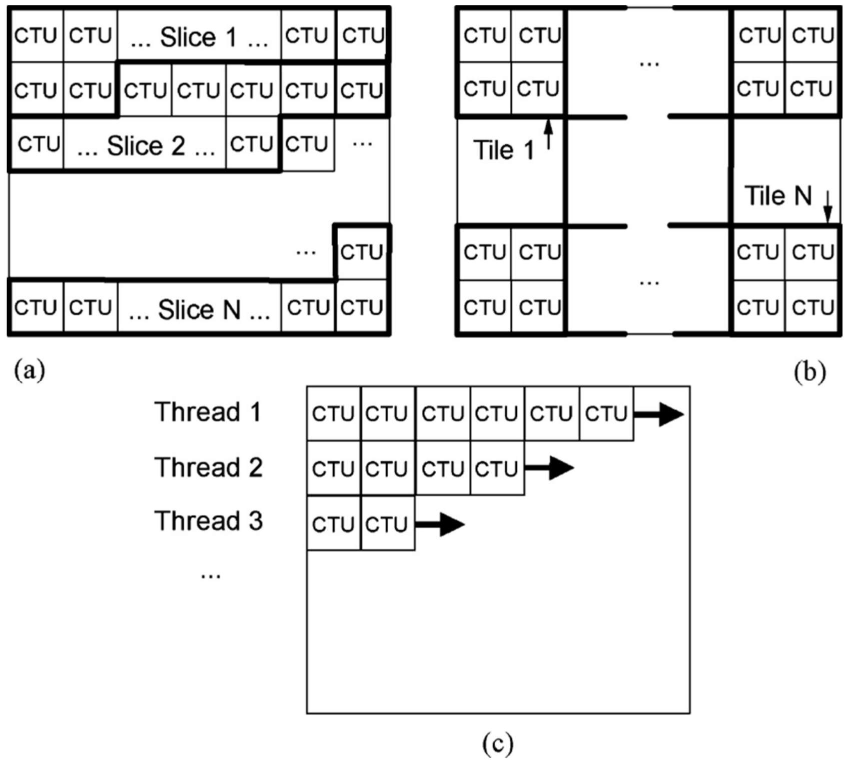


Figure 2.6: Subdivision of picture into a) slices and b) tiles. c) Wavefront parallel processing [6]

2.1.9 Intra Picture Prediction

Intra coding is the coding of current block using previously coded blocks in the same image. The previously coded parts form the prediction basis for current block. This takes into account the spatial correlation between neighboring blocks. After the prediction is made, only the difference is coded, which in turn leads to improved coding efficiency.

2.2 Intra Picture Prediction in H.264/AVC

For the purpose of luma intra prediction in H.264 for a 4x4 block, there are nine intra prediction modes provided [8]. One of these modes is the DC mode. The other eight modes provide angular prediction. These directional modes include – vertical, horizontal, diagonal down-left, diagonal down-right, vertical-right, horizontal-down, vertical-left and horizontal-up [8]. The vertical mode uses pixels directly above the current block for prediction. Similarly the horizontal mode uses pixels directly to the left of current block. All other directional modes are at approximately equal angles. These are shown in Figure 2.7.

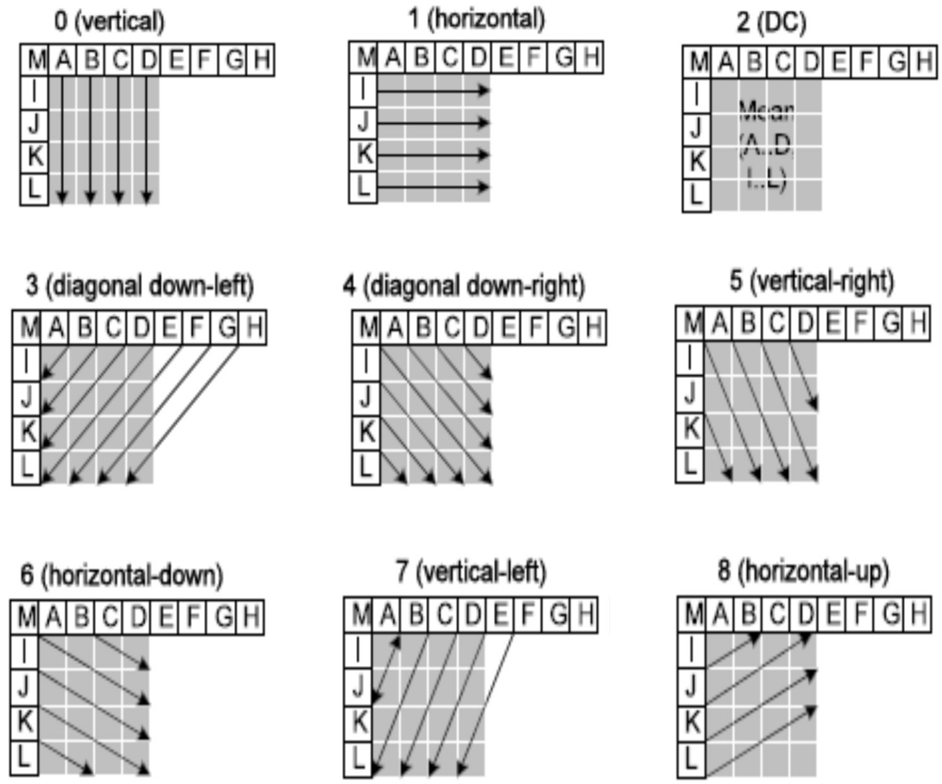
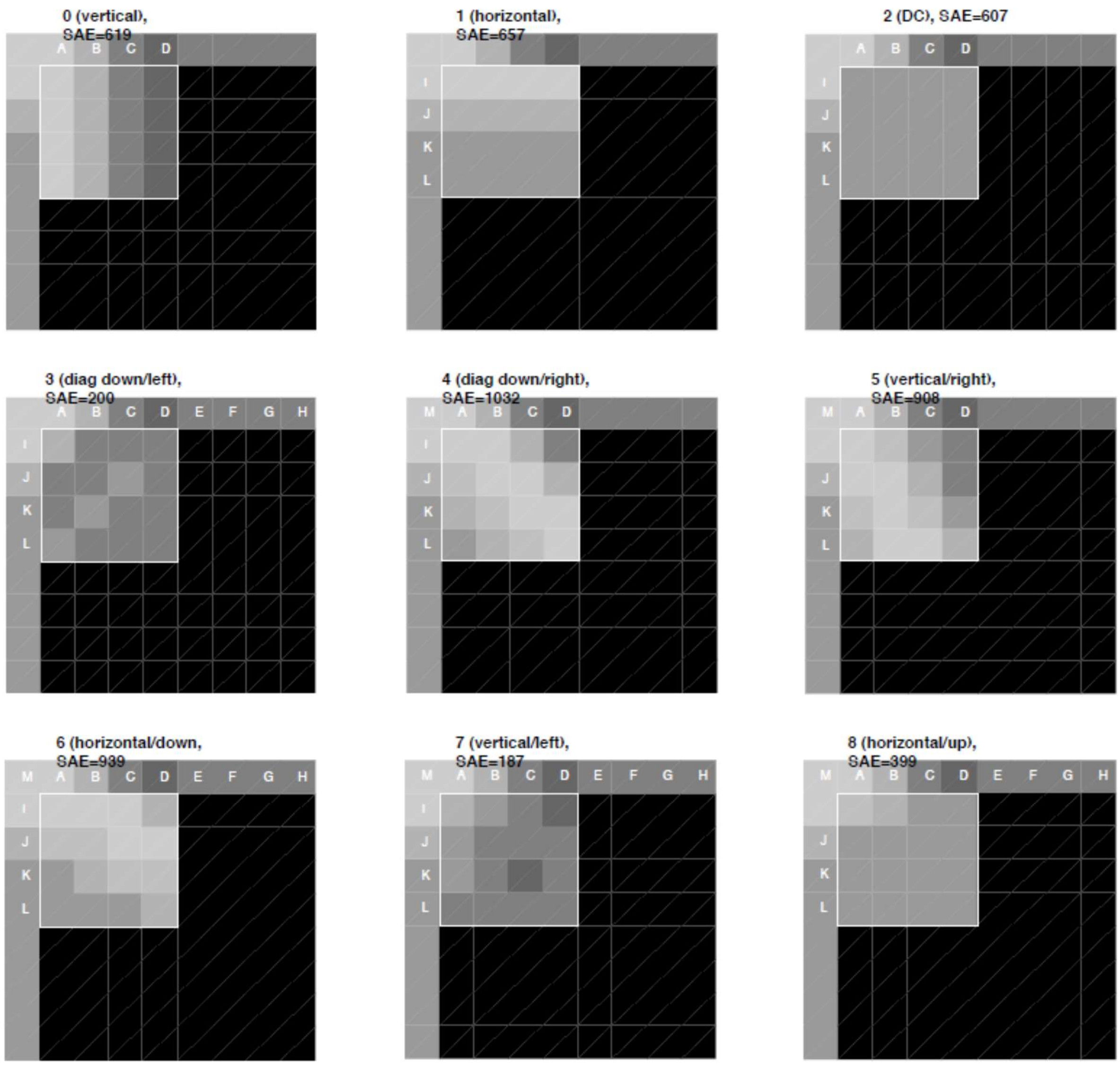


Figure 2.7: Nine luma prediction modes of H.264 [7]

Angular intra prediction or directional prediction uses pixels in different directions depending on the content. Although having multiple angular directions for prediction improves coding efficiency, it makes the encoding algorithm more complex. Hence it is a tradeoff between the quality desired and the complexity affordable.

As can be seen from the image above, the previously coded samples which are above and left to the current block are used for predicting the current block. The encoder decided which prediction mode to be used based on the residual value. The residual value is represented in terms of Sum of Absolute Errors (SAE). The mode which provided least value of SAE is chosen for the current block [7]. The Figure 2.8 shows an example of prediction blocks and the decision made.

Intra prediction for 16x16 blocks supports only four modes – *vertical, horizontal, DC and Plane* [7].



Firgure 2.8: Prediction block decision [7]

Most Probable Mode

After the prediction is done, the mode which has been chosen must be coded along with the residual information. Since there are nine modes, coding them as it is will become inefficient. For the purpose of coding these intra prediction modes, H.264 supports one most probable mode (MPM). This method considers the fact that the chosen mode is often same as what was chosen for previously coded blocks. Using this information the mode which has highest chances of being the current intra prediction mode is chosen as MPM [8]. If the current mode indeed turns out to be the same as the one predicted, only a single bit is sufficient to code the current mode. Otherwise, the remaining eight modes are coded and sent to the decoder [8].

2.3 Intra Picture Prediction in HEVC

HEVC takes the intra prediction techniques of H.264 forward. It introduced thirty-five intra prediction modes to improve directional prediction accuracy and coding efficiency. Thirty-three of these are angular prediction modes and the rest two are DC mode (or flat mode) and Planar mode (or surface fitting mode) [6]. For chroma 5 modes are available which includes – DC mode, planar mode, horizontal mode, vertical mode and a direct copy of luma intra prediction mode. Figure 2.9 shows all directional modes for H.264 (left) and HEVC (right).

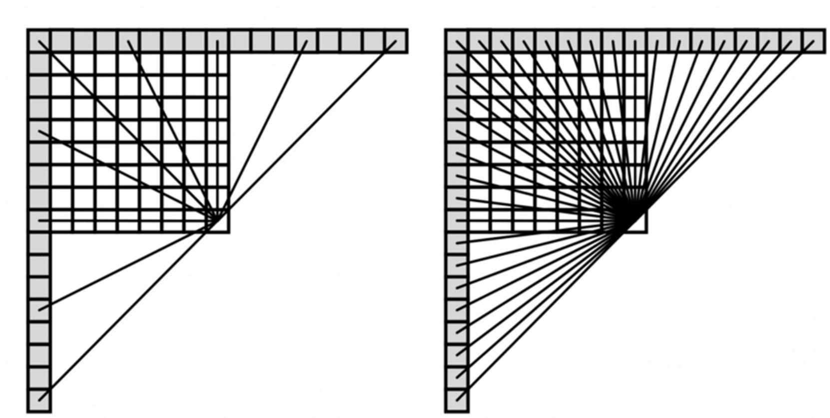


Figure 2.9: Intra prediction directions of H.264 (left) and HEVC (right) [10]

DC Mode

DC mode uses the average values of pixels from the reference pixels above and left of the current PU [6].

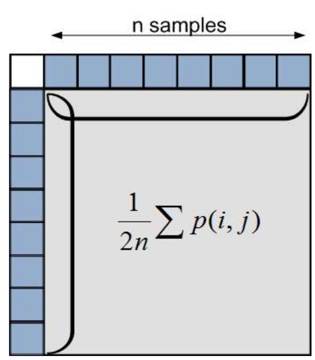


Figure 2.10: DC mode in intra prediction [11]

Planar Mode

Planar prediction is used to predict texture area more efficiently [12]. This mode is used to prevent discontinuities and have a smoother texture along block boundaries [6].

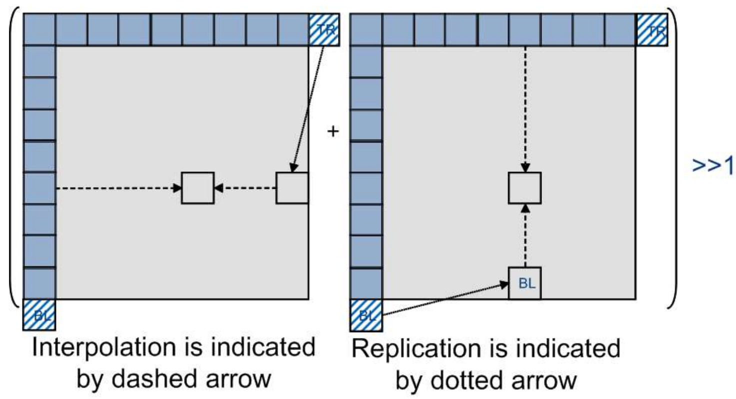


Figure 2.11: Planar mode prediction and interpolation [11]

HEVC supports mode dependent intra smoothing (**MDIS**). The reference pixels used in intra prediction are filtered to reduce the high frequencies. HEVC uses a three-tap filter for this process. The filtering decision depends on the prediction direction and PU size, among other factors. This smoothing decision is based on the RD (Rate Distortion) cost [12].

2.4 Intra Mode Coding in HEVC

During the course of the development of the HEVC standard, a number of methods were proposed for intra mode coding. The method [2] which was adopted in the standard divides the thirty-five intra prediction modes into two categories - three most probable modes and thirty-two remaining modes. The three MPMs are derived considering the modes of the PU which are to the left and above of the current PU. If the current intra prediction mode is same as one of the three MPMs, its index is coded, using only one or two bits. If the current mode is among the remaining modes, a 5-bit fixed length code is used to signal the mode. This method [2], thus, uses less bits for transmitting modes with a higher probability and more bits for modes which have a lower probability. As compared to the single MPM method used in H.264/AVC [8], this method provided 1.0% BD rate savings when tested for All Intra configuration [2].

2.5 Intra picture prediction in JEM

Since high-resolution video content are becoming more popular, more number of intra prediction modes are required for accurate prediction and improved coding efficiency. The first version of the reference software - JEM1.0 – defined sixty-seven intra prediction modes [15]. The three MPM based method of HEVC was modified to six MPM based method in JEM to take into account the extended intra prediction modes [15]. As such, these modes formed two categories – six most probable modes, and sixty-one remaining non-MPM modes. The method of coding these intra prediction modes was kept similar to the method used in HEVC. If the current intra prediction mode was one among the six selected MPMs, its index was coded. Otherwise a fixed length code was used to signal one of the remaining modes [15].

2.6 MPM derivation method in JEM

The method JEM employs [29] to derive the six most probable modes uses five neighbors of the current PU. These five neighbors are defines as :

Left, Above, Above left, Below left, Above right

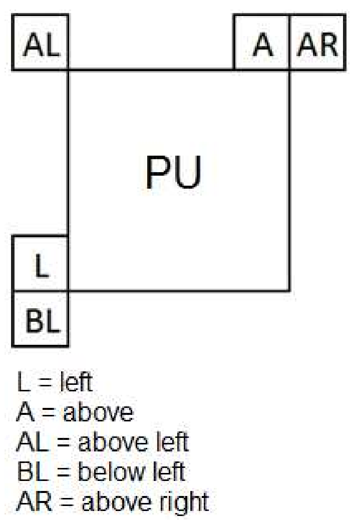


Figure 2.12: Neighbors of PU used in MPM modes derivation [29]

Using these neighbors the following order is used for deriving the six MPMs.

left, above, planar (0), DC (1), below left, above right, above left,
left -1, left +1, above -1, above +1, below left -1, below left +1, above right -1, above right +1, above left -1, above left +1,
vertical, horizontal, 2, diagonal

The first six available and unique modes are chosen to be the most probable modes.

2.7 Non-MPM mode coding in JEM

With the increasing number of intra prediction modes, more number of bits are required to code them. As such, reducing the number of bits for intra mode coding is of high priority. In the second JVET meeting in February 2016, a method [19] was proposed to further divide the remaining sixty-one non-MPM modes into two groups – the selected modes group and the non-selected modes group. According to this method [19], sixteen selected modes are extracted from these non-MPM modes, leaving remaining forty-five modes. The leftover forty-five modes fall into the non-selected category. Before this method was adopted, all remaining sixty-one modes were coded using a fixed length code. This method changed that. Now, a search is performed to check whether the current intra prediction mode is among the sixteen selected modes. If found, it is signaled using a 4-bit fixed length code. If it is among the non-selected modes, a truncated binary code is used to signal the current mode. This method was adopted in the JEM test model [20].

The method described above [19] works as follows. After the six MPMs have been extracted, the remaining sixty-one modes are sorted and rearranged to have index from zero to sixty.

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 …58, 59, and 60}

These are divided into two sets. The first set is obtained by selecting every fourth mode, and is called the selected modes set. The remaining modes form non-selected modes set. There is a flag in the bit stream which indicates the category the current mode belongs to. The index of the selected modes set and non-selected modes set is given below:

- Selected modes - {0, 4, 8, 12, 16, 20, …60}
- Non-selected modes – {1, 2, 3, 5, 6, 7, 9, …58, 59}

The problem with this approach is that the criterion for deriving the selected modes is not based on probability of occurrence. Every forth mode in the sorted list is extracted to be in selected category. Since modes falling in selected category are coded using a shorter code, it makes sense to select those modes which have a higher probability. Further, the forty-five remaining modes are coded using truncated binary coding, which effectively further divides them into two smaller groups. The first part of these non-selected modes group contain nineteen modes, and would be coded using a 5-bit code. The second part of these modes contain the last twenty-six modes. These would be coded using a 6-bit code.

Considering these facts, a better method to code these intra prediction modes would be to arrange them as per their probability of occurrence and derive modes falling in each category accordingly. This paper proposes a new method which takes into consideration this fact. Probability statistics generated offline are used to rearrange all intra prediction modes. This order is used to make a decision about the category the current intra prediction modes fall into, and as such the number of bits that would be required to code this mode.

The coding of non-MPM modes remained unchanged in next several versions of test model [20][21][22][23][24]. There have been, however, proposals that deal with non-MPM mode coding. The next section reviews and discusses some of these proposed methods for non-MPM mode coding. This includes extracting the selected modes and non-selected modes.

2.6 Non-MPM Derivation Method in G0060

A new method to derive the selected modes set was proposed at the seventh JVET meeting in Geneva [25]. This method considers all of the angular modes among the six most probable mode list. Since DC mode and Planar mode are always present in the MPM list, there would be exactly four angular modes in this list. These angular modes are then used to derive modes to be added in the selected mode category.

The derivation process is done by adding and subtracting an offset of 2 from these modes. When the list is exhausted and sixteen unique modes have not been found, the process continues with the modes in selected category as input. This iterative process continues till 16 unique angular modes are obtained. An example of this is given below.

MPM set	{0, 1, 50, 18, 2, 34}
Selected modes set	{48, 52, 16, 20, 65, 4, 32, 36, 46, 54, 14, 22, 63, 6, 30, 38}
Non-selected modes set	Others

Table 2.2: Example of selected modes set using method in G0060 [25]

Using the example in the above table, angular modes 50, 18, 2, 34 from the MPM list are considered. The selected modes obtained by taking an offset of 2 iteratively would be: {48, 52, 16, 20, 65, 4, 32, 36, 46, 54, 14, 22, 63, 6, 30, 38}. This method used JEM6.0 as an anchor and resulted in an overall 0.1% BD rate savings [25].

2.7 Non-MPM Derivation Method in H0029

At the eighth JVET meeting, a new method [26] was proposed to derive 16 selected modes from the remaining 61 non-MPM modes to improve coding efficiency.

In this method [26], the angular modes in the MPM list are considered and derivation of selected modes is done by adding and subtracting offsets of 1, 2, 3 and 4 to these angular modes. The number of derived modes obtained w.r.t to MPM index is given by (4,3,3,2,2,2). This means that the first mode in the MPM is used to derive plus 4 and minus 4 (total 8) modes, the second is used to derive plus 3 and minus 3 (total 6) modes, and so on. Only unique modes are added to the selected category and the process is carried out until 16 modes are obtained.

If even after this process, 16 unique modes are not obtained, default modes are added similar to 6 MPM derivation. This default order is defined as {2, 18, 34, 50, 66; 10, 26, 42, 58; 6, 14, 22, 30, 38, 46, 54, 62; 17, 19, 49, 51}.

This method uses JEM7.0 as anchor and achieved 0.1% luma BD rate savings [26].

3

3. Probability Based Intra Mode Coding

3.1 Probability based MPM mode coding

This method focusses on deriving the six MPMs based on their probability of occurrence. As discussed before, JEM3.0 defines a predefined order for deriving the six MPMs [29]. This order is

left, above, planar (0), DC (1), below left, above right, above left,
left -1, left +1, above -1, above +1, below left -1, below left +1, above right -1, above right +1, above left -1, above left +1,
vertical, horizontal, 2, diagonal

Our method derives this ordering based on their probability of occurrence. The probability statistics are derived from offline processing of test streams. A total of 24 test streams with four QP values each (22, 27, 32, 37) are used for generating a frequency table. This frequency table consist of all possible intra prediction modes (0 to 66) as rows and all possible candidate modes as columns. The candidates are the defined as the five neighbors, their adjacent modes upto an offset of +/- 5, along with six constant modes.

L, A, BL, AR, AL,
L-1, L+1, A-1, A+1, BL-1, BL+1, AR-1, AR+1, AL-1, AL+1,
L-2, L+2, A-2, A+2, BL-2, BL+2, AR-2, AR+2, AL-2, AL+2,
L-3, L+3, A-3, A+3, BL-3, BL+3, AR-3, AR+3, AL-3, AL+3,
L-4, L+4, A-4, A+4, BL-4, BL+4, AR-4, AR+4, AL-4, AL+4,
L-5, L+5, A-5, A+5, BL-5, BL+5, AR-5, AR+5, AL-5, AL+5,
0, 1, 50, 18, 2, 34

Where L = left, A = above, BL = below left, AR = above right, AL = above left

For each PU, if the current intra prediction mode is same as that of one of the candidates, that particular point in the 2D table is incremented. After this process is done for all 96 streams (24 streams with 4 QP values each), the generated table would provide the overall frequency of each candidate for all streams. These candidates are then sorted to obtain a candidate order.

	B	C	D	E	F	G	H
1	**L**	**A**	**AL**	**0**	**1**	**AR**	**BL**
2	8048279	6458261	6200343	18490074	0	3476008	2849254
3	5283526	4104478	3876016	0	12267447	2010797	1807459
4	518890	372960	300475	0	0	137829	154920
5	202251	154844	128007	0	0	63752	67057
6	103401	77369	58117	0	0	26814	32254
7	94660	68678	51506	0	0	24409	25825
8	135844	91379	65713	0	0	29196	34542
9	124117	89982	67989	0	0	33078	37043
10	119965	82279	65420	0	0	32098	35810

Figure 3.1: Partial snapshot of the frequency table generated.

The candidate order obtained from this table from highest to lowest frequency is:

L, A, AL, 0, 1, AR, BL, 50, A-1, AL-1, L-1, 18, 2, A+1, AL+1, L+1, AR-1, AR+1, BL+1, BL-1, A-2, AL+2, A+2, AL-2, L+2, 34, L-2, A-3, AL+3, AL-3, A+3, L+3, AR-2, AR+2, A-4, AL+4, A+4, AL-4, L+4, L-3, A-5, A+5, AL+5, AL-5, L-4, L+5, AR-3, BL+2, AR+3, L-5, BL-2, AR-4, AR+4, AR-5, BL+3, AR+5, BL-3, BL+4, BL-4, BL+5, BL-5

Six unique modes obtained from this order go into the MPM list.

The results for this method, which was ran for class D streams using JEM3.0 as an anchor are provided below.

Class	Stream	BD rate (AI)
D	BasketballPass	0.09%
	BQSquare	0.03%
	BlowingBubbles	0.00%
	RaceHorses	0.08%

Table 3.1: Test results for probability based MPM list derivation for class D streams using JEM3.0 as anchor

As can be seen from the results, this method not seem to be working.

3.2 Dynamically derive first MPM

Out of all the six MPMs, the first one is the most important one, since signaling the first MPM requires only one bit. JEM3.0 always uses left mode as the first one and above as the second (if both are available). However, from the frequency table generated earlier, we can see that the probability of occurrence of above mode can be higher than that of the left mode.

	B	C	D	E	F	G	H
1	L	A	AL	0	1	AR	BL
2	8048279	6458261	6200343	18490074	0	3476008	2849254
3	5283526	4104478	3876016	0	12267447	2010797	1807459
4	518890	372960	300475	0	0	137829	154920
5	202251	154844	128007	0	0	63752	67057
6	103401	77369	58117	0	0	26814	32254
7	94660	68678	51506	0	0	24409	25825
8	135844	91379	65713	0	0	29196	34542
9	124117	89982	67989	0	0	33078	37043
10	119965	82279	65420	0	0	32098	35810

Figure 3.2: Comparing probability of occurrence of L (left) and A (above) modes.

As can be seen from the figure above, for some values of L and A, probability of occurrence of A can be higher than that of L. Using this information, the order of the first and the second MPM can be changed. For each PU, if both left and above neighbors are available, their frequencies are compared. If the frequency of above mode is found to be higher than that of left mode, the order is swapped and above is made the first MPM.

The table below shows test results for class C, D and E streams using JEM3.0 as anchor. This method improved performance for some streams but the results are not consistently good.

Class	Stream	BD rate (AI)
C	BasketballDrill	-0.03%
	BQMall	-0.04%
	PartyScene	0.01
	RacehorsesC	0.03%
D	BasketballPass	0.08%
	BQSquare	0.01%
	BlowingBubbles	-0.01%
	RaceHorses	0.04%
E	FourPeople	0.00%
	Johnny	0.15%
	KristenAndSara	-0.03%

Table 3.2: Test results for probability based first MPM selection using JEM3.0 as anchor.

3.3 Joint probability method for first MPM selection

This method uses joint probability of left and above neighbors to make a decision about the first MPM. For generating statistics, 24 test streams with 4 QP values each, are processed offline. A 2D matrix with above mode as column and left mode as row is created to calculate probability of (left, above) as a pair. For each PU (in all 96 streams), if both left and above modes are available, and the current intra prediction mode is same as that if the left mode, matrix1 is incremented at that (left, above) point. Similarly, for each PU if both left and above neighbors are available and the current intra prediction mode is same as that of the above mode, that (left, above) point is incremented in a second table, matrix2.

This provides us with two matrices. Matrix1 contain the probability of left mode w.r.t above mode. Matrix2 contain the probability of above mode w.r.t left mode. A third matrix is generated by subtracting matrix1 from matrix2. This residual matrix would contain information where the probability of above mode is higher than that of the left mode. All the points where this residual matrix has positive values are the points where above has a higher probability. Similarly all the points which have negative values are the points where probability of left mode is higher. As such, we disregard the negative values, considering only the positive ones. For all the pairs of (left, above) where the values are positive, the order of left and above modes is swapped to make above as the first MPM.

The results for this method are provided in Table 3.3. As can be seen, this method improved performance for most streams, but still does not work for some.

Class	Stream	BD rate (AI)
C	BasketballDrill	-0.04%
	BQMall	-0.01%
	PartyScene	-0.02%
	RacehorsesC	-0.01%
D	BasketballPass	0.06%
	BQSquare	0.00%
	BlowingBubbles	-0.02%
	RaceHorses	0.03%
E	FourPeople	-0.04%
	Johnny	0.26%
	KristenAndSara	-0.08%

Table 3.3: Test results for joint probability method using JEM3.0 as anchor.

3.4 Probability based non-MPM mode coding

The method (G0060) [25] described previously derives selected modes by taking offsets from the angular modes in the most probable mode list. These offsets start from 2 and increase in a step size of 2, considering both positive and negative values. The reasoning behind this method is not clear and it does not take into account probabilities of occurrence of all the modes.

Similarly, the method (H0029) [26] calculates selected modes by taking offsets of ±1, ±2, ±3 and ±4 to the angular modes in the MPM list. The offsets w.r.t the MPM index is given by {4,3,3,2,2,2}. Although the implementation of this method is simple, this order is not based on their probabilities.

To solve this problem a method is proposed to arrange and order intra prediction modes based on their probabilities of occurrence. This method generates a probability matrix by processing test streams offline. A total of 24 streams with 4 QP (Quantization Parameter) values (22, 27, 32, 37) were used for this processing. Each column of this matrix is a candidate mode, while each row is a different intra prediction mode (from 0 to 60). Each cell [row, column] in this matrix contain the number of times a candidate (column) is an angular mode and is not present in the most probable mode list. The total occurrence of each candidate mode is then observed and the candidate list is sorted from most occurring candidate to least occurring candidate. Hence, this matrix gives the probabilities of a candidate being a non-MPM mode. This offline data, a candidate mode order, is used to make a decision how the current modes is coded. Since our method uses JEM7.0 [27] as an anchor, the method to detect non-MPM modes used in offline processing is the same as the one used in JEM7.0.

3.4.1 Method using six MPM modes

The candidates in this method are obtained from the six MPM modes. These six MPM modes are used to derive their adjacent modes, which form the candidate mode list. These include offsets of ±1, ±2, ±3, ±4 and ±5 to these MPM modes. This gives ten neighbors for each mode, resulting in a total of sixty candidate modes. Let the six MPMs be named as

MPM0, MPM1, MPM2, MPM3, MPM4, MPM5

The candidates are then defined as:

- MPM0+1, MPM0-1, MPM0+2, MPM0-2, MPM0+3, MPM0-3, MPM0+4, MPM0-4, MPM0+5, MPM0-5

- MPM1+1, MPM1-1, MPM1+2, MPM1-2, MPM1+3, MPM1-3, MPM1+4, MPM1-4, MPM1+5, MPM1-5
- MPM2+1, MPM2-1, MPM2+2, MPM2-2, MPM2+3, MPM2-3, MPM2+4, MPM2-4, MPM2+5, MPM2-5
- MPM3+1, MPM3-1, MPM3+2, MPM3-2, MPM3+3, MPM3-3, MPM3+4, MPM3-4, MPM3+5, MPM3-5
- MPM4+1, MPM4-1, MPM4+2, MPM4-2, MPM4+3, MPM4-3, MPM4+4, MPM4-4, MPM4+5, MPM4-5
- MPM5+1, MPM5-1, MPM5+2, MPM5-2, MPM5+3, MPM5-3, MPM5+4, MPM5-4, MPM5+5, MPM5-5

The candidate order as obtained after offline processing, from highest to lowest probability of occurrence is given by:

mpm0_m1, mpm0_a1, mpm1_m1, mpm1_a1, mpm0_a2, mpm0_m2, mpm1_m2, mpm1_a2, mpm0_a3, mpm0_a4, mpm1_m3, mpm0_m3, mpm1_a3, mpm1_m4, mpm0_m4, mpm0_a5, mpm1_a4, mpm1_m5, mpm1_a5, mpm0_m5, mpm3_a1, mpm3_m1, mpm2_m1, mpm3_a2, mpm3_m2, mpm4_a2, mpm4_a1, mpm4_m2, mpm4_m1, mpm3_a3, mpm3_m3, mpm2_a1, mpm4_m3, mpm3_a4, mpm4_a3, mpm3_m4, mpm3_m5, mpm3_a5, mpm4_a4, mpm4_m4, mpm4_m5, mpm4_a5, mpm2_a4, mpm2_m3, mpm2_a2, mpm2_a3, mpm2_m4, mpm2_m2, mpm2_a5, mpm2_m5, mpm5_m1, mpm5_m2, mpm5_a1, mpm5_a2, mpm5_m3, mpm5_a3, mpm5_a4, mpm5_m4, mpm5_a5, mpm5_m5

Figure 3.3 shows a partial snapshot of the probability table which has been sorted to display highest to lowest probabilities.

	A	B	C	D	E	F	G	H
1	**mpm0_m1**	**mpm0_a1**	**mpm1_m1**	**mpm1_a1**	**mpm0_a2**	**mpm0_m2**	**mpm1_m2**	**mpm1_a2**
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	34135	93410	15454	45608	22553	9279	6822	15555
5	14674	33363	6255	20361	29393	3225	2443	16586
6	10473	16673	4612	5940	12004	3052	2420	7877
7	11235	8468	5766	3721	6471	2934	2078	2898
8	11288	5141	5452	2101	2034	4213	2839	1338
9	13616	7118	4976	4287	2661	3969	2645	1879
10	14554	10400	6235	3730	3346	7219	4634	2438
11	42861	8568	13229	3588	4550	10388	6236	2743

Figure 3.3: Snapshot of probability table using neighbors of 6 MPMs.

For each PU, after the six MPMs have been extracted, sixteen unique modes are obtained according to this order. If the sixteen modes required for selected category modes are not obtained after this order has been exhausted, a default mode order is used. This default order is given as follows:

{50, 66, 2, 18, 54, 62, 58, 10, 14, 6, 22, 46, 45, 26, 34, 42}

If the current luma intra prediction mode is among these sixteen selected modes, a 4-bit fixed length code is used to signal the mode.

Results for this method in Table 3.4 below shows good performance over class E streams. Tests were performed for All Intra using JEM7.0 as anchor.

Class	Stream	All Intra Main 10 Over JEM7.0		
E	FourPeople	-0.19%	-0.29%	-0.08%
	Johnny	-0.23%	-0.42%	-0.41%
	KristenAndSara	-0.31%	-0.28%	-0.31%

Table 3.4: Test results for probability based non-MPM coding using 6 MPM method.

3.4.2 Method using four MPM angular modes

The candidates in this method are derived from the angular modes in the six MPM list. Since DC mode and Planar mode are always present in the MPM list, there would be exactly four angular modes in the list. The candidates are offsets of these angular modes. Offset of ±1, ±2, ±3, ±4 and ±5 are extracted, which results in ten neighbors per angular mode. This gives a total of forty candidates. Let the angular modes in the MPM list be called

- MPM0, MPM1, MPM2 and MPM3

The forty candidates are then named as:

- MPM0+1, MPM0-1, MPM0+2, MPM0-2, MPM0+3, MPM0-3, MPM0+4, MPM0-4, MPM0+5, MPM0-5
- MPM1+1, MPM1-1, MPM1+2, MPM1-2, MPM1+3, MPM1-3, MPM1+4, MPM1-4, MPM1+5, MPM1-5
- MPM2+1, MPM2-1, MPM2+2, MPM2-2, MPM2+3, MPM2-3, MPM2+4, MPM2-4, MPM2+5, MPM2-5
- MPM3+1, MPM3-1, MPM3+2, MPM3-2, MPM3+3, MPM3-3, MPM3+4, MPM3-4, MPM3+5, MPM3-5

The order of these candidate modes, arranged from highest probability to lowest probability as obtained from statistics is as follows:

MPM0-1, MPM0+1, MPM0+2, MPM0-2, MPM1-1, MPM1+1, MPM0+3, MPM1-2, MPM1+2, MPM0+4, MPM0-3, MPM0-4, MPM0+5, MPM1-3, MPM0-5, MPM1+3, MPM1-4, MPM1+4, MPM1-5, MPM1+5, MPM2+2, MPM2+1, MPM2-2, MPM2-1,

MPM2-3, MPM2+3, MPM2+4, MPM2-4, MPM2-5, MPM2+5, MPM3-1, MPM3-2, MPM3+1, MPM3+2, MPM3-3, MPM3+3, MPM3+4, MPM3-4, MPM3+5, MPM-5

Figure 3.4 shows a snapshot of the probability table generated using these forty candidates.

	A	B	C	D	E	F	G	H
1	**mpm0_m1**	**mpm0_a1**	**mpm0_a2**	**mpm0_m2**	**mpm1_m1**	**mpm1_a1**	**mpm0_a3**	**mpm1_m2**
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	44511	122711	26620	11210	14892	41021	15861	9227
5	19566	48729	36884	3836	4794	20421	5782	3665
6	14403	20588	16975	3867	3294	5804	13610	3276
7	17160	10702	8006	3560	2911	3380	9138	2829
8	16586	6899	2441	5167	3701	1936	3048	3655
9	17936	10134	3162	4649	3520	4425	2705	3530
10	19410	13686	4111	8235	5296	3345	3164	6225

Figure 3.4: Probability table generated using four angular mode method

Using this pre-available order, the sixteen selected modes are obtained similar to the previous method. The default mode order, to be used if sixteen unique modes are not obtained after this list is exhausted, are the same as in the previous method. This order is:

{50, 66, 2, 18, 54, 62, 58, 10, 14, 6, 22, 46, 45, 26, 34, 42}

If the current intra prediction mode is one of these sixteen selected modes, a 4-bit fixed length code is used to signaling.

The test results for this method for class E streams over JEM7.0 are given below.

Class	**Stream**	**All Intra Main 10 Over JEM7.0**		
E	FourPeople	-0.19%	-0.29%	-0.08%
	Johnny	-0.23%	-0.42%	-0.41%
	KristenAndSara	-0.31%	-0.28%	-0.31%

Table 3.5: Test results for probability based non-MPM coding using 4 angular MPM method.

It can be inferred from the above results that the four angular MPM method works better than the six MPM method of previous section. Below are the test results for class B, C, D and E streams. The method works consistently good for all these streams.

Class	All Intra Main 10 Over JEM7.0		
	Y	U	V
B	-0.08%	-0.11%	-0.08%
C	-0.14%	-0.10%	0.00%
D	-0.12%	-0.11%	0.04%
E	-0.27%	-0.28%	-0.17%

Table 3.6: Test results for four angular MPM method for class B, C, D and E streams.

3.5 Extending four angular MPM method to order all 67 intra modes

The sixty-seven intra prediction modes consist of six MPMs, sixteen selected modes and forty-five non-selected modes [19]. In JEM7.0, the forty-five modes in the non-selected category are coded using truncated binary coding. As such, they are divided into two groups. The first part of consist of nineteen modes and they are signaled using 5-bit fixed length coding (FLC). The second part consist of twenty-six modes which are signaled using 6-bit fixed length coding.

67 modes = 6 MPMs + 16 selected + 19 non-selected (part 1) + 26 non-selected (part 2)

Using this information, our method arranges the intra prediction modes in an array according to probability order derived above. This is done by generating an array consisting of all sixty-seven intra prediction modes. The first six modes in this array are the most probable modes which have already been derived. The method to derive the first 6 most probable modes and their signaling remains unchanged from JEM7.0. After the six MPMs have been derived, a total of thirty-five (16+19) modes are chosen based on the pre-defined order obtained from probability statistics. The first sixteen form the selected modes set and next nineteen form the first part of the non-selected modes group. The remaining twenty-six modes are the second part of the non-selected modes group and fill up the end of the array. Modes in the selected category are signaled using 4-bit fixed length code. Modes falling in the non-MPM group of first nineteen modes are coded using 5-bit fixed length coding. Finally the remaining last twenty-six are signaled using 6-bit fixed length coding. This is depicted in the Table 3.7.

The algorithm for intra mode coding works as follows:

- If current intra prediction mode is among 6 MPM
    - Mode signaled using unary code
- Else
    - If current mode is among next 16 selected modes
        - Mode signaled using 4-bit FLC

- Else if current mode is among next 19 modes
  - Mode signaled using 5-bit FLC
- Else (current mode is among last 26 modes)
  - Mode signaled using 6-bit FLC

Intra prediction modes	**Code**
First 6 modes	Unary code
Next 16 modes (selected)	4-bit FLC
Next 19 modes (non-selected first part)	5-bit FLC
Next 26 modes (non-selected second part)	6-bit FLC

Table 3.7: Proposed method for Intra mode coding

Entropy coding uses shorter codes for information which has higher probability, and longer codes for information having lower probability. Our method takes advantage of this particular theory and as such result in BD rate savings.

3.6 Experimental Results for proposed method

At the second JVET meeting in San Diego in February 2016, the common test conditions (CTC) were released [28]. The methods discussed in this report has been tested using these conditions for All Intra configuration and JEM7.0 [27] as an anchor. For generating offline statistical data and for testing, twenty-four test video streams were used with four QP (quantization parameter) values (22, 27, 32 and 37).

Offline processing of these streams generated a table of probability data of previously described forty candidates. This was then sorted to generate an order of candidates from highest probability to lowest probability.

The Table 3.8 below shows the experimental results of our proposed method for All Intra (AI) configuration. As compared to JEM7.0, it shows a BD rate improvement of 0.13%.

Class	All Intra Main 10 Over JEM7.0		
	Y	U	V
A1	-0.10%	-0.06%	-0.10%
A2	-0.10%	-0.07%	-0.14%
B	-0.07%	-0.08%	-0.12%
C	-0.14%	-0.15%	-0.11%
D	-0.16%	0.04%	0.03%
E	-0.28%	-0.36%	-0.17%
Overall	**-0.13%**	**-0.10%**	**-0.10%**

Table 3.8: BD rate performance of the proposed method using JEM7.0 as anchor

Table 3.9 shows results for each stream. As can be observed, class E streams show the most improvement, an average of 0.28% BD rate savings. Among class E streams, the stream “KristenAndSara” shows highest BD rate reduction of 0.34%. Class B, on the other hand, shows least improvement with an average value of 0.07% BD rate reduction.

Class	Stream	All Intra Main 10 Over JEM7.0		
		Y	U	V
A1	Tango2	-0.17%	-0.06%	-0.12%
	Drums100	-0.14%	-0.02%	-0.07%
	Campfire	-0.07%	-0.09%	-0.18%
	ToddlerFountain2	-0.04%	-0.08%	-0.03%
A2	CatRobot	-0.17%	-0.01%	-0.11%
	TrafficFlow	-0.05%	-0.21%	-0.11%
	DaylightRoad2	-0.07%	-0.03%	-0.27%
	Rollercoaser2	-0.09%	-0.01%	-0.06%
B	Kimono	-0.04%	-0.01%	0.04%
	ParkScene	0.00%	-0.02%	-0.05%
	Cactus	-0.11%	-0.16%	-0.23%
	BasketballDrive	-0.16%	-0.13%	-0.07%
	BQTerrace	-0.03%	-0.10%	-0.27%
C	BasketballDrill	-0.20%	-0.19%	-0.14%
	BQMall	-0.12%	-0.22%	-0.19%
	PartyScene	-0.06%	-0.09%	-0.03%
	RaceHorses	-0.18%	-0.09%	-0.09%
D	BasketballPass	-0.16%	0.10%	0.09%
	BQSquare	-0.04%	0.16%	0.23%
	BlowingBubbles	-0.19%	-0.08%	-0.21%
	RaceHorses	-0.24%	0.00%	0.01%

E	FourPeople	-0.24%	-0.36%	-0.09%
	Johnny	-0.27%	-0.35%	0.04%
	KristenAndSara	-0.34%	-0.36%	-0.47%
Overall		-0.13%	-0.10%	-0.10%

Table 3.9: Per stream BD rate performance of proposed method

4

4. Conclusion

The objective of this research was to study and explore new methods for coding intra prediction modes for luma, so as to reduce the BD rate and improve overall coding efficiency. The research explored a number of new methods for coding the MPM and non-MPM modes in intra prediction. These methods are based on probability statistics derived offline. The experimental results suggest that this does not work very well for MPM modes. However, non-MPM mode coding shows considerable improvement.

According to the method proposed for non-MPM coding, the four angular intra prediction modes among the six MPMs are extracted. These are then used to derive a total of forty candidate modes. The probabilities of these candidate modes are generated and this information is used to select modes that fall into selected and non-selected category among the non-MPM modes. The idea is to code the modes with a higher probability with fewer bits compared to modes with a lower probability, which is in line with the principle of entropy coding.

When compared to JEM7.0 software, this method shows average luma BD rate savings of 0.13%.

References

[1] Cisco Visual Networking Index forecast 2017, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

[2] Ximin Zhang, Shan Liu, Shawmin Lei, "Intra mode coding in HEVC standard", Visual Communications and Image Processing (VCIP), 2012 IEEE.

[3] http://www.emmys.com/video/69th-engineering-emmy-awards-joint-collaborative-team-video-coding-wins-emmy-award

[4] http://news.itu.int/itu-iso-iec-receive-another-primetime-emmy-for-video-compression-video/

[5] https://jvet.hhi.fraunhofer.de/

[6] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 12, pp. 1649–1668, 2012.

[7] Iain Richardson, White paper : "H.264/AVC Intra Prediction", Vcodex.

[8] T. Wiegand; G. J. Sullivan; G. Bjontegaard; A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560-576, 2003.

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

[10] Shohei Matsuo, Seishi Takamura and Atsushi Shimizu, "Modification of Intra Angular Prediction in HEVC", Proceedings of the 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference, 2012, Pages1-4.

[11] O le meur, "Video Compression Beyond H.264 – HEVC", Univ. of Rennes 1, Nov 7, 2011.

[12] Nam Ling, Lecture notes – COEN 338, SCU, Mar 2017.

[13] N. Kamaci, Y. Altunbasak, "Performance comparison of the emerging H.264 video coding standard with the existing standards", Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on,

[14] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards including high efficiency video coding (HEVC)," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 12, pp. 1669–1684, 2012.

[15] J. Chen, E. Alshina, G.-J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of Joint Exploration Test Model 1," JVET-A1001, Geneva, Oct. 2015.

[16] Rickard Sjoberg, Ying Chen, Akira Fujibayashi, Miska M. Hannuksela, Jonatan Samuelsson, Thiow Keng Tan, Ye-Kui Wang, Stephan Wenger , "Overview of HEVC High-Level Syntax and Reference Picture Management", IEEE Transactions on Circuits and Systems for Video Technology, Year: 2012, Volume: 22, Issue: 12, Pages: 1858 – 1870.

[17] Philipp Helle, Simon Oudin, Benjamin Bross, Detlev Marpe, M. Oguz Bici, Kemal Ugur, Joel Jung, Gordon Clare, and Thomas Wiegand, "Block Merging for Quadtree-Based Partitioning in HEVC", IEEE Transactions on Circuits and Systems for Video Technology, 2012, vol 22, Issue 12, Pages 1720-1731.

[18] Wikipedia, High Efficiency Video Coding, https://en.wikipedia.org/wiki/High_Efficiency_Video_Coding

[19] S.-Hwan Kim and A. Segall, "Further improvement of intra coding tools", JVET-B0051, San Diego, USA, Feb. 2016.

[20] J. Chen, E. Alshina, G.-J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of Joint Exploration Test Model 3," JVET-C1001, Geneva, Jun. 2016.

[21] J. Chen, E. Alshina, G.-J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of Joint Exploration Test Model 4," JVET-D1001, Chengdu, Oct. 2016.

[22] J. Chen, E. Alshina, G.-J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of Joint Exploration Test Model 5," JVET-E1001, Geneva, Jan. 2017.

[23] J. Chen, E. Alshina, G.-J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of Joint Exploration Test Model 6," JVET-F1001, Hobart, Mar. 2017.

[24] J. Chen, E. Alshina, G.-J. Sullivan, J.-R. Ohm, and J. Boyce, "Algorithm description of Joint Exploration Test Model 7," JVET-G1001, Torino, Jul. 2017.

[25] Y. Han, J. An, J. Zheng, "Improvements for Intra Prediction Mode Coding", JVET-G0060, Torino, Jul. 2017.

[26] V. Seregin, W.-J. Chien, M. Karczewicz, N. Hu, X. Zhao, "EE1: Secondary MPM list", JVET-H0029, Macau, Oct. 2017.

[27] https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-7.0/

[28] K. Suehring, X. Li, "JVET common test conditions and software reference configurations", JVET-B1010, San Diego, USA, Feb. 2016.

[29] Vadim Seregin, Xin Zhao, Amir Said, and Marta Karczewicz, "Neighbor based intra most probable modes list derivation", Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Doc. JVET-C0055, 4th Meeting.

Acronyms

VNI	Visual Network Index
IP	Internet Protocol
VoD	Video on Demand
VR	Virtual Reality
CDN	Content Delivery Network
AVC	Advanced Video Coding
HDTV	High Definition Television
SDTV	Standard Definition Television
JCT-VC	Joint Collaborative Team on Video Coding
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
VCEG	Video Coding Expert Group
MPEG	Moving Pictures Expert Group
HEVC	High Efficiency Video Coding
UHC	Ultra High Definition
JVET	Joint Video Exploration Team
JEM	Joint Exploration Model
MPM	Most Probable Mode
NAL	Network Abstraction Layer
VPS	Video Parameter Set
SPS	Sequence Parameter Set
PPS	Picture Parameter Set
VCL	Video Coding Layer
AUD	Access Unit Delimiter
EOS	End Of Sequence
EOB	End Of Bitstream
FD	Filler Data
SEI	Supplemental Enhancement Information
GOP	Group Of Pictures
VUI	Video Usability Information
MV	Motion Vectors
CTU	Coding Tree Unit
CTB	Coding Tree Block
CU	Coding Unit
CB	Coding Block
PU	Prediction Unit

PB	Prediction Block
TU	Transform Unit
TB	Transform Block
SAE	Sum of Absolute Errors
MDIS	Mode Dependent Intra Smoothing
RD	Rate Distortion
QP	Quantization Parameter
FLC	Fixed Length Coding
AI	All Intra